

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LE ROUTAGE DANS LES RÉSEAUX AD HOC À RADIOS COGNITIVES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR

LAMIA ELGAROUÏ

MAI 2013

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

« Celui qui ne remercie pas les gens, ne remercie pas son créateur »

Je remercie mon directeur de recherche Mr Wessam Ajib pour m'avoir offert son encadrement et son savoir.

Je tiens à remercier le grand homme qui m'a soutenu et encouragé durant toute ma vie mon cher père Boubekur, aussi je remercie celle qui m'a ensevelie par sa tendresse et son amour ma très chère mère Kilali Khedaoudj.

Les durs moments vécus durant mon acheminement académique loin de mes trois perles précieuses Sana, Sérine et Saoussene, m'ont poussé à tenir bon afin d'achever ce travail et de le leur offrir pour qu'elles soient fières de leur mère.

Je n'oublie pas de remercier toutes les personnes que j'ai côtoyées à l'université du Québec à Montréal, mes collègues du laboratoire TRIME et mes amies Hasna Boudra, Soumia Belaidouni, ainsi que tous les professeurs et le personnel administratif que j'ai eu la chance de connaître durant ma formation au département informatique.

Aussi, je remercie celles qui m'ont appris comment surmonter les moments difficiles de la vie en savourant chaque instant de mon existence, à vous Karima Belkahla et Amina Faci.

À tous merci.

TABLE DES MATIÈRES

LISTE DES FIGURES	vii
LISTE DES TABLEAUX	xi
LISTE DES ABRÉVIATIONS	xiii
RÉSUMÉ.....	xvii
CHAPITRE I	
INTRODUCTION ET GÉNÉRALITÉS	1
1.1 Introduction	1
1.2 Définitions.....	2
1.2.1 Usager primaire et usager secondaire	2
1.2.2 Protocoles de routage pour les réseaux à radios cognitives	2
1.2.3 Les radios cognitives.....	4
1.3 Présentation de réseaux ad hoc à radios cognitives.....	4
1.4 Avantages des RRCs et motivation de leur utilisation	5
1.5 Description du problème de routage dans les RRCs ad hoc.....	6
1.5.1 Catégorie 1 : environnement statique.....	6
1.5.2 Catégorie 2 : environnement dynamique	7
1.5.3 Catégorie 3 : environnement opportuniste.....	7
CHAPITRE II	
MÉTHODOLOGIE ET APPROCHES	9
2.1 Présentation de la problématique	9
2.2 Revue de littérature	10
2.3 Synthèse de l'état de l'art.....	20
CHAPITRE III	
CONTRIBUTION	23
3.1 Présentation du protocole de routage réactif TORA	23
3.2 Caractéristiques et avantages	28

3.3	Hypothèses associées à notre proposition	28
3.4	Description de la solution proposée.....	29
CHAPITRE IV		
MISE EN ŒUVRE DE LA SOLUTION PROPOSÉE ET SIMULATION.....		31
4.1	Présentation de l'outil de simulation NS	31
4.2	Mise en œuvre de la solution.....	32
4.2.1	Modification du protocole TORA au concept multi-interface	32
4.2.2	Adaptation du protocole TORA aux RRCs	37
4.3	Simulation et évaluation des performances	40
4.3.1	Paramètres de simulation	40
4.3.2	Métriques de performance.....	44
CHAPITRE V		
RÉSULTATS ET INTERPRÉTATION.....		45
5.1	Interprétation des résultats.....	45
5.1.1	Première étape : évaluation de TORA multi-interface.....	45
5.1.2	Deuxième étape : adaptation au réseau à radios cognitives.....	71
CHAPITRE VI		
CONCLUSION ET PERSPECTIVES.....		77
6.1	Conclusion.....	77
6.2	Perspectives.....	78
BIBLIOGRAPHIE.....		81

LISTE DES FIGURES

Figure	Page
3.1 Modélisation de chemins par graphe acyclique orienté dans le protocole de routage TORA (V. D. Park et M. S. Corson 1997, 2).....	24
3.2 Processus de création de route (V. D. Park et M. S. Corson 1997, 2).....	26
4.1 Architecture d'un nœud mobile dans NS-2 (R. A. Calvo et J. P. Campo, 2007)	33
4.2 Architecture d'un nœud mobile multi-interface adaptée par (R. A. Calvo et J. P. Campo, 2007 ; P. Lee et G. Wei 2009)	34
4.3 Interaction du protocole TORA avec le protocole IMEP (V. D. Park et M. S. Corson 1999)	36
4.4 Format d'un paquet QRY (<i>QueRY</i>) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)	37
4.5 Format d'un paquet UPD (<i>UPDate</i>) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)	38
4.6 Format d'un paquet CLR (<i>CLEAR</i>) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)	39
5.1 Variation du débit moyen en fonction de la vitesse de mobilité, trafic TCP	46
5.2 Variation du débit moyen en fonction de la vitesse de mobilité, trafic CBR	47
5.3 Variation du délai de bout en bout en fonction de la vitesse de mobilité, trafic TCP	48
5.4 Variation du délai de bout en bout en fonction de la vitesse de mobilité, trafic CBR	49
5.5 Variation du nombre moyen de sauts en fonction de la vitesse de mobilité, trafic TCP	50
5.6 Variation du nombre moyen de sauts en fonction de la vitesse de mobilité, trafic CBR	51
5.7 Variation de la charge normalisée en fonction de la vitesse de mobilité, trafic TCP	52
5.8 Variation de la charge normalisée en fonction de la vitesse de mobilité, trafic CBR	53

5.9	Variation de la charge de routage en fonction de la vitesse de mobilité, trafic TCP	54
5.10	Variation de la charge en fonction de la vitesse de mobilité, trafic CBR	55
5.11	Variation du taux de livraison normalisé en fonction de la vitesse de mobilité, trafic TCP	56
5.12	Variation du taux de livraison normalisé en fonction de la vitesse de mobilité, trafic CBR	57
5.13	Variation du taux de pertes en fonction de la vitesse de mobilité, trafic CBR	58
5.14	Variation du débit moyen en fonction de temps de pause des nœuds, trafic TCP	59
5.15	Variation du débit moyen en fonction de temps de pause des nœuds, trafic CBR	60
5.16	Variation du délai de bout en bout en fonction de temps de pause des nœuds, trafic TCP	61
5.17	Variation du délai de bout en bout en fonction de temps de pause des nœuds, trafic CBR	62
5.18	Variation du nombre moyen de sauts en fonction de temps de pause des nœuds, trafic TCP	63
5.19	Variation du nombre moyen de sauts en fonction de temps de pause des nœuds, trafic CBR	64
5.20	Variation de la charge normalisée en fonction de temps de pause des nœuds, trafic TCP	65
5.21	Variation de la charge du réseau en fonction de temps de pause des nœuds, trafic CBR	66
5.22	Variation du PDR en fonction de temps de pause des nœuds, trafic TCP	67
5.23	Variation du taux de livraison normalisé (PDR) en fonction de temps de pause des nœuds, trafic CBR	68
5.24	Variation du taux de pertes en fonction de temps de pause des nœuds, trafic TCP	69
5.25	Variation du taux de pertes en fonction de temps de pause des nœuds, trafic CBR	70
5.26	Processus de découverte et d'établissement de chemin entre nœuds secondaires ..	72
5.27	Présence de nœuds primaires en zone d'activité de nœuds secondaires	73

5.28	Transfert de données entre nœuds primaires et cessation de communication entre nœuds secondaires.	74
5.29	Variation du débit entre nœuds secondaires en présence d'activité de nœuds primaires	75

LISTE DES TABLEAUX

Tableau	Page
4.1 Tableau des paramètres de simulation	41

LISTE DES ABRÉVIATIONS

AODV	Protocole de routage vecteur de distance sur demande (en anglais, <i>Ad hoc On demand Distance Vector</i>)
ARP	Protocole de résolution d'adresse (en anglais, <i>address resolution protocol</i>)
CAODV	Protocole <i>AODV Cognitive</i>
CBR	Taux de transmission constant (en anglais, <i>Constant bit rate</i>)
CLR	Paquet de destruction de chemin (en anglais, <i>CLeaR</i>)
CRP	Protocole de routage cognitif (en anglais, <i>Cognitive Routing Protocol</i>)
CR	Radio cognitive (en anglais, <i>Cognitive radio</i>)
CTBR	Protocole de routage cognitif à base d'arbre (en anglais, <i>Cognitive Tree-based Routing</i>)
DAG	Graphe acyclique dirigé (en anglais, <i>Directed Acyclic Graph</i>)
DSDV	Protocole de routage vecteur de distance par séquence de destination (en anglais, <i>Destination Sequenced Distance Vector</i>)
DSR	Protocole de routage source dynamique (en anglais, <i>Dynamique Source Routing</i>)
ETT	Temps de transmission prévu (en anglais, <i>Expected Transmission Time</i>)
ETX	Compteur de transmission prévu (en anglais, <i>Expected Transmission Count</i>)

FCC	Commission Fédérale de Communication (en anglais, <i>Federal Communication Commission</i>)
FFSR	(En anglais, <i>First Fast Second Reliable</i>)
FNR	Protocole de routage du voisin le plus proche (en anglais, <i>Farthest-Neighbor Routing</i>)
FTP	Protocole de transfert de fichiers (en anglais, <i>File Transfer Protocol</i>)
IMEP	Protocole d'encapsulation pour Internet (en anglais, <i>Internet MANET Encapsulation Protocol</i>)
IZR	Zone de routage indépendante (en anglais, <i>Independent Zone Routing</i>)
MAC	Couche d'accès au médium (en anglais, <i>medium access control</i>)
MANET	Réseaux mobile ad hoc (en anglais, <i>Mobile ad hoc networks</i>)
MBP	Métrique basée sur les probabilités (en anglais, <i>Metric Based on Probabilities</i>)
MILP	Programmation linéaire mixte en nombres entiers (en anglais, <i>Mixed Integer Linear Programming</i>)
MINLP	Programmation non linéaire mixte en nombre entiers (en anglais, <i>Mixed Integer Non-Linear Programming</i>)
MISF	Fonction de sélection multi-interface (en anglais, <i>Mutli-Interface Intelligent Sélection Function</i>)
MPSF	Fonction de sélection multi-chemin (en anglais, <i>Multi-Path Selection Function</i>)

NNR	Protocole de routage du voisin le plus proche (en anglais, <i>Nearest-Neighbor Routing</i>)
NS	Simulateur de réseaux (en anglais, <i>Network Simulator</i>)
OFDM	Multiplexage par répartition de fréquence orthogonales (en anglais, <i>Orthogonal Frequency Division Multiplexing</i>)
OLSR	Protocole de routage d'optimisation d'état de lien (en anglais, <i>Optimized Link State Routing</i>)
OPT	Paquet d'optimisation (en anglais, <i>OPTimized</i>)
OTCL	Langage de programmation TCL orienté objet
PCTC	Protocole de contrôle de topologie et de routage cognitif (en anglais, <i>Protocol Cognitive Topology Control and routing in Mobile Ad hoc</i>)
PU	Usager primaire (en anglais, <i>primary user</i>)
QRY	Paquet de requête (en anglais, <i>QueRY</i>)
ROSA	Protocole de routage distribué et allocation dynamique du spectre dans les réseaux à radios cognitives (en anglais, <i>Distributed Joint Routing and Dynamique Spectrum Allocation in Cognitive Radio Ad hoc Networks</i>)
RRC	Réseau à radios cognitives
RREP	Paquet de réponse de transmission (en anglais, <i>Route REPLY</i>)
RREQ	Paquet de requête d'établissement de route (en anglais, <i>Route REQuest</i>)
SAMER	(En anglais, <i>Spectrum Aware Mesh Routing</i>)

SARP	(En anglais, <i>Spectrum Aware Routing Protocol for cognitive ad hoc networks</i>)
SEARCH	(En anglais, <i>SpEctrum Aware Routing for Cognitive ad hoc networks</i>)
SPEAR	(En anglais, <i>SpEctrum Aware Routing protocole</i>)
STOD-RP	(En anglais, <i>Spectrum-Tree based On-Demand Routing Protocol</i>)
SU	Usager secondaire (en anglais, <i>Secondary User</i>)
TBR	Routage à base d'arbre (en anglais, <i>Tree Based Routing</i>)
TCP	Protocole de contrôle de transmission (en anglais, <i>Transmission Control Protocol</i>)
TORA	Protocole de routage d'ordonnancement temporaire (en anglais, <i>Temporary Ordering Routing Algorithm</i>)
UC	Usager cognitif
WECTT	(En anglais, <i>Weighted Cumulative expected Transmission Count</i>)
ZRP	Protocole de routage par zone (en anglais, <i>Zone Routing Protocol</i>)

RÉSUMÉ

Les réseaux à radios cognitives sont des réseaux émergents donnant une possibilité d'accès dynamique au spectre dans le but de l'utiliser de façon plus efficace. Le problème de routage dans ce type de réseaux est assez complexe vu l'aspect dynamique et flexible d'un environnement cognitif opportuniste. Notre travail consiste à suggérer une solution de routage efficace et performante en débit et délai de transmission de bout en bout. Dans ce mémoire, nous proposons un nouveau protocole de routage réactif en s'inspirant des protocoles de routage déjà existant dans les réseaux sans fil classiques car ils présentent une grande maturité. Les simulations effectuées démontrent que l'algorithme permet d'atteindre un débit de transmission élevé avec un petit délai de transmission de bout en bout, ainsi qu'un bon maintien de chemins face à l'activation de transmission d'utilisateurs primaires.

Mots clés : réseaux à radios cognitives, routage réactif, multi-interface, utilisateur primaire, utilisateur secondaire.

CHAPITRE I

INTRODUCTION ET GÉNÉRALITÉS

1.1 Introduction

La communication filaire a évolué, laissant place à une communication sans fil qui répond mieux à la forte demande et aux exigences des usagers, en particulier en ce qui concerne la mobilité. De plus, les réseaux sans fil sont devenus très populaires depuis qu'ils sont liés au réseau mondial Internet. Particulièrement, les réseaux sans fil ad hoc ont pris une grande importance chez les usagers grâce au faible coût de leurs équipements, leur forte capacité de liaison et leur simple configuration.

La présence de plusieurs réseaux sans fil a eu pour conséquence le partage du spectre en différentes bandes. Chaque bande est subdivisée en canaux et chaque canal est alloué à un seul usager donné. L'allocation de cette ressource précieuse (c.-à-d. le spectre) devient très difficile. La saturation du spectre et sa non disponibilité pour des futurs usagers engendrent un problème complexe. Devant cette situation, une étude minutieuse a révélé que l'occupation du spectre n'est pas à 100% à tout moment (Khalife et al. 2009), et que cette ressource est mal exploitée car durant la non-utilisation de certains canaux attribués aux usagers avec licence, d'autres usagers sans licence ne peuvent pas en bénéficier. En revanche, il est nécessaire de respecter vigoureusement la priorité des usagers avec licence.

De plus, la partie du spectre sous licence est utilisée de façon sporadique. Cette variation d'utilisation de ces portions oscille de 15% à 85% dans le temps (Khalife et al. 2009), donnant lieu à des espaces blancs ou des trous de spectre. L'exploitation de ces trous est possible via une nouvelle génération d'équipements intelligents et de cartes radios

programmables qui sont capables d'éliminer l'interférence, d'apprendre de l'état du canal, de contrôler l'interférence et d'accéder dynamiquement au spectre.

Ainsi de nouveaux réseaux sont apparus, ils ont été nommés réseaux à radios cognitives (RRCs) car ils permettent, de façon intelligente, de détecter la disponibilité des canaux et l'établissement de liaison. Ce type de réseaux intelligents est confronté à beaucoup de contraintes, à savoir, la puissance de transmission, le routage, l'interférence et la qualité de service. Dans le cadre de notre projet, nous étudierons de près le problème du routage dans les réseaux ad hoc, notamment dans les réseaux ad hoc à radios cognitives.

1.2 Définitions

Plusieurs nouveaux termes et nouvelles notions liées aux réseaux RRCs sont récemment apparues. Nous utiliserons l'abréviation de ces termes afin d'alléger le texte.

1.2.1 Usager primaire et usager secondaire

L'utilisateur primaire et l'utilisateur secondaire sont deux termes utilisés pour désigner un utilisateur privilégié et un utilisateur non privilégié respectivement. Les utilisateurs primaires sont des usagers ayant le droit d'utilisation du canal de communication à tout moment car ils détiennent une licence d'utilisation, tandis que les utilisateurs secondaires n'ont pas ce privilège et doivent reporter leur communication jusqu'à ce que le canal soit disponible.

1.2.2 Protocoles de routage pour les réseaux à radios cognitives

Ces protocoles cherchent un chemin qui permettra une connexion entre une source et une destination. Pour trouver ce chemin de routage, il faut choisir un protocole de routage adéquat afin d'assurer une route offrant de hautes performances pour une connexion secondaire efficace sans que cette connexion ne soit interrompue par un nœud primaire. Le cas échéant, il faut trouver un autre chemin de routage sans déranger les usagers primaires (Yang Xu et al. 2010).

Dans les réseaux ad hoc traditionnels, il existe principalement trois types de protocoles de routage nommés : proactifs, réactifs et hybrides.

Les protocoles proactifs. Dans ce type de protocoles, il faut établir les routes avant la demande d'une connexion entre une source et une destination. Cela incite à chercher tous les chemins d'un nœud donné vers tous les nœuds du réseau s'il en existe et de sauvegarder ces chemins dans une table avec toutes les informations appropriées (temps de découverte, état de la liaison, etc.). La maintenance des chemins choisis se fait périodiquement par des échanges de données de contrôle. Ce type de protocole est loin d'être approprié aux réseaux ad hoc à radios cognitives vu la nécessité d'avoir préalablement l'information des routes à suivre alors que cette information dépend étroitement des usagers primaires. Il se peut que toutes les routes découvertes à un instant donné soient non valides à un autre instant dû à l'appariation d'une connexion primaire entre les deux instants. Donc, il en résultera une nouvelle découverte de route et beaucoup de perte de temps pour l'établissement de connexion et lancement de trafic. Parmi les protocoles les plus répandus dans cette catégorie citons : DSDV (en anglais *Destination Sequenced Distance Vector*) et OLSR (en anglais *Optimized Link State Routing*).

Les protocoles réactifs. Ces protocoles se basent sur le principe de découverte de route à la demande. Quand une source veut envoyer des données vers une destination, un mécanisme de découverte de route sera lancé en premier. Une fois le chemin découvert, il sera possible d'établir la connexion et lancer le transfert des données. La maintenance de route n'est pas périodique comme dans le cas des protocoles proactifs, afin d'économiser les ressources du réseau. Ce type de protocoles répond mieux aux exigences des réseaux à radios cognitives, car les nœuds secondaires qui ont besoin de communiquer font une demande de connexion à chaque fois afin d'exploiter les bandes de spectre non utilisés par les nœuds primaires. Toutefois, il faut les adapter en ajoutant une prise en charge des contraintes de présence de nœuds primaires et des réactions à entreprendre dans ces cas. Le protocole AODV (en anglais *Ad hoc On demand Distance Vector*) est un de ces protocoles. Il a fait l'objet de plusieurs études, et a été souvent proposé comme le protocole de routage pour les RRCs. Il est à signaler que les protocoles réactifs ont des caractéristiques qui peuvent convenir aux RRCs.

Les protocoles hybrides. Le mot hybride définit déjà ce type de protocole qui essaie de jumeler les caractéristiques des deux approches définies précédemment (protocoles réactifs et protocoles proactifs), dans le but d'éviter leurs inconvénients et d'exploiter leurs avantages. Au niveau de chaque nœud, l'information concernant les voisins est sauvegardée, tandis que l'établissement de chemin se fait à la demande. Parmi ces protocoles nous trouvons : ZRP (en anglais *Zone Routing Protocol*) et IZR (en anglais *Independent Zone Routing*).

1.2.3 Les radios cognitives

Les utilisateurs secondaires sont équipés de cartes à radios cognitives. Ils doivent coexister d'une manière transparente avec les utilisateurs primaires. Les RCs peuvent exploiter les bandes sous licence soit pendant l'absence des usagers primaires ou par le calcul judicieux de leur puissance de transmission afin de bénéficier de la partie sous-utilisée du spectre. Les deux stratégies doivent être réalisées avec soin, en évitant tout impact négatif sur les utilisateurs primaires.

1.3 Présentation de réseaux ad hoc à radios cognitives

Les réseaux ad hoc à radios cognitives sont des réseaux de télécommunication sans fil munis d'une capacité cognitive afin d'utiliser efficacement la ressource radio. Ils suivent les cycles (étapes) cognitifs suivants (Ngom et Diouf 2008) :

- 1- Étape d'apprentissage et d'analyse de l'environnement (connaître les canaux occupés, calculer les puissances maximales d'émission, et mesurer le débit).
- 2- Étape d'identification des actions à entreprendre afin de répondre aux besoins selon la situation en cours de l'environnement.
- 3- Étape de prise de décision où il faut savoir quoi faire et comment agir vis-à-vis de l'état actuel de l'environnement.

De ces faits, on peut dire que les réseaux ad hoc à radios cognitives doivent être capables de découvrir et d'adapter les besoins des nœuds qui les composent, ainsi que de

gérer les ressources spectre et énergie de façon adéquate selon la priorité attribuée à ces nœuds. Donc, les réseaux ad hoc à radios cognitives sont des réseaux qui peuvent examiner leur environnement, l'analyser et sauvegarder en mémoire l'état de cet environnement pour faciliter l'action en conséquence.

1.4 Avantages des RRCs et motivation de leur utilisation

Sur la base de ce qui a été dit sur les réseaux ad hoc à radios cognitives, on peut voir plusieurs avantages pratiques. Ces avantages nous motivent à proposer ces réseaux pour les situations où le spectre est mal exploité. Les avantages les plus significatifs que nous pouvons lister sont:

- accès flexible au spectre optimisant ainsi la performance du système;
- coexistence de réseaux hétérogènes primaires-secondaires en garantissant un fonctionnement sans dégrader les performances des réseaux primaires;
- amélioration de l'exploitation du spectre en donnant la possibilité à tout usager de communiquer quelle que soit la disponibilité de canal;
- exécution de plusieurs fonctions radios sur un même système matériel;
- amélioration des protocoles de gestion et de routage dans les RRCs en ajoutant des fonctions qui permettent d'assurer un fonctionnement continu et de garantir une qualité de service;
- apparition de nouveaux concepts de fonctionnement inter-couches permettant l'échange d'information entre les couches du réseau sur la disponibilité du spectre et
- transmission parallèle des données réduisant le temps de transmission.

Ces avantages nous poussent plus à nous pencher sur le fonctionnement de ce nouveau type de réseau et de voir de près les problèmes les plus complexes qui sont non résolus, tel que le routage.

1.5 Description du problème de routage dans les RRCs ad hoc

La découverte d'un chemin entre deux utilisateurs secondaires et l'établissement d'une connexion qui ne sera pas interrompue par l'apparition d'un utilisateur primaire jusqu'à la fin de la communication n'est pas toujours garantie. Donc, le protocole de routage à adopter pour les RRCs doit être capable de trouver le chemin de routage où la communication peut se faire, sans coupure, et le cas échéant, il faut rapidement résoudre cette coupure par un autre chemin de routage afin que la communication puisse se compléter.

La conception d'un protocole de routage pour les réseaux ad hoc à radios cognitives est une problématique importante pour laquelle une solution adéquate qui répond aux cas de présence ou d'absence d'utilisateurs primaires, n'altérant pas la communication, n'a toujours pas été trouvée, mais des propositions ont été faites. Dans les différentes études faites jusqu'à présent plusieurs solutions ont été proposées où chacune a ses propres contraintes (Venkaiahnaidu et al. 2012).

En pratique, il n'y a pas de solution générale pour le routage dans les RRCs qui peut être proposée dû à l'environnement cognitif qui peut être classé en trois catégories distinctes. Chaque catégorie nécessite des solutions de routage propres.

1.5.1 Catégorie 1 : environnement statique

Sur une bande primaire relativement statique, où la durée de l'inactivité des nœuds primaires dépasse largement la durée de communication des utilisateurs secondaires (on les appelle aussi les utilisateurs cognitifs), la solution de routage peut être utilisée comme dans les réseaux ad hoc classiques (Khalife et al. 2009). D'un point de vue utilisateur cognitif, la bande de fréquence disponible peut être exploitée pour une durée illimitée et là nous pouvons employer un des protocoles de routage réactifs existants. De plus, ces protocoles ont atteint une certaine maturité de développement et de stabilité.

1.5.2 Catégorie 2 : environnement dynamique

Lorsque les bandes de fréquences disponibles sont plus dynamiques c'est-à-dire que l'occupation du spectre par les utilisateurs primaires est plus fréquente, une nouvelle solution de routage spécifique doit être proposée. La bande primaire peut être exploitée par un utilisateur cognitif, mais de façon discontinue, ce qui affecte gravement la qualité de service offerte pour une communication secondaire (Khalife et al. 2009).

1.5.3 Catégorie 3 : environnement opportuniste

Dans ce cas, les utilisateurs secondaires essayent d'accéder à des bandes de fréquences dont la disponibilité est rare et très aléatoire. La solution optimale est d'assurer les ressources radios pour les transmissions cognitives tout en garantissant le service pour tous les utilisateurs primaires en communication sur les canaux exploités sur l'ensemble du chemin. La découverte appropriée de chemin du nœud source vers le nœud destination dans une topologie qui évolue de manière dynamique peut être un problème très difficile à résoudre. C'est un cas très dynamique. Les utilisateurs primaires utilisent plus souvent leurs bandes de fréquence. La disponibilité de bandes de fréquences pour toute la durée de communication secondaire devient très difficile. Une solution possible mais très complexe à implémenter, pour les nœuds cognitifs, est de transmettre de façon opportuniste sur les bandes de fréquence disponibles pendant la courte période de leur disponibilité (Khalifé et al. 2009).

1.6 Organisation du mémoire

Ce mémoire est organisé comme suit. Dans le premier chapitre intitulé introduction et généralités, nous avons présenté les réseaux ad hoc à radios cognitives, leurs avantages et la motivation de leur utilisation, ainsi qu'une description du problème de routage lié aux RRCs. Dans le second chapitre, nous illustrons les approches les plus récentes dans la littérature proposant des solutions au problème de routage dans les RRCs, suivi d'une synthèse de toutes les solutions présentées dans cette étude. Ensuite dans un troisième chapitre, nous exposons notre contribution pour résoudre le problème de routage dans les RRCs. Nous présentons le protocole de routage classique TORA, son fonctionnement, ses caractéristiques et ses avantages. Puis, nous construisons des hypothèses pour notre solution afin de répondre aux exigences des RRCs. La mise en œuvre de notre solution a été traitée dans un quatrième

chapitre. Une brève présentation de l'outil de simulation NS-2 est faite ainsi qu'une description des modifications à apporter au protocole de routage TORA afin de l'adapter aux RRCs. L'évaluation des performances du protocole est faite par simulation en utilisant le simulateur NS-2. L'évaluation se fait en fonction de certaines métriques importantes telles que le débit, le délai de bout en bout, le taux de pertes, etc. Dans un cinquième chapitre, une interprétation des résultats obtenus est donnée, en deux étapes. La première étape est consacrée aux modifications du protocole TORA multi-interface et une seconde étape est pour l'adaptation de ce dernier aux RRCs. Enfin, un sixième chapitre présente la conclusion tirée de ce travail, avec d'éventuelles perspectives de travaux futurs.

CHAPITRE II

MÉTHODOLOGIE ET APPROCHES

2.1 Présentation de la problématique

La radio cognitive est une nouvelle technologie visant à résoudre le problème des portions non-occupées (dites trous) du spectre et garantir une meilleure utilisation de cette ressource. Cette technologie émergente a donné naissance à de nouveaux réseaux utilisant des composants matériels capables de changer leur configuration selon l'environnement spectral. Cette capacité d'adaptation a ouvert de nouvelles avenues de recherche pour la conception de stratégies d'accès au support avec comportement opportuniste pour la réutilisation des trous de spectre temporairement vacants. Il n'est pas simple d'appliquer cette technique aux différentes couches du réseau, ce qui amène une grande complexité dans la conception de nouveaux protocoles répondant aux défis de ces nouveaux RRCs. En plus d'assurer les fonctionnalités définies dans les réseaux sans fil classiques, il est aussi nécessaire de gérer le changement de l'environnement spectral lié à l'activité des usagers primaires. De ce fait, l'interconnexion des usagers cognitifs en coexistence avec des usagers primaires nous met devant un problème complexe au niveau de la couche réseau vu l'hétérogénéité du nouveau système. Le choix de bande de fréquence de transmission allouée à un nœud doit être fait selon la disponibilité et la capacité de celle-ci. En conséquence, le routage dans les RRCs est basé sur des techniques inter-couches (la couche physique, la couche MAC et la couche réseau) et doit exploiter la transmission sur différentes bandes de fréquence selon leur disponibilité.

En se basant sur le chapitre précédent, les solutions trouvées jusqu'à présent imposent des contraintes sur le matériel ou sur la méthode d'établissement du chemin de routage. La revue de littérature associée à notre projet nous a permis de découvrir plusieurs propositions de solutions, que nous résumons dans ce qui suit.

2.2 Revue de littérature

Dans (Cesana et al. 2010), la solution de routage proposée prend en considération, soit une connaissance générale du spectre, soit une connaissance locale. Pour la connaissance générale du spectre, une carte d'occupation du spectre est disponible pour tous les nœuds du réseau. Elle est représentée par une base de données du spectre centralisée. Cette base de données est maintenue régulièrement par la commission fédérale de communication FCC (Khalife et al. 2009). Elle a pour but de stocker l'information concernant le temps et l'espace de disponibilité de canal. Pour ce faire, les méthodes suivies sont basées sur des principes de graphes où le réseau est modélisé par des nœuds et des arrêtes, ou sur une méthode d'optimisation.

Dans le cas de modélisation par des graphes, la méthode est composée de deux phases. La première phase est l'abstraction de graphe qui se réfère à la génération d'un graphe logique représentant la topologie physique du réseau $G = (N, V, f(V))$ où N désigne le nombre de nœuds, V désigne le nombre de liens, $f(V)$ est la fonction allouant un poids à chaque lien. Ce poids est attribué selon la disposition du lien. Si c'est un lien horizontal, alors le poids sera alloué en fonction de la bande passante ou en fonction du changement de lien, si c'est un lien vertical, alors le poids sera alloué en fonction du coût de changement de fréquence. La seconde phase permet de construire un chemin dans le graphe reliant les paires source-destination. Cela revient à relier le nœud source et le nœud destination à des nœuds intermédiaires par des arêtes. Sur chaque arête, un poids est alloué par la fonction $f(V)$. Ce poids représente la métrique du protocole de routage.

Deux types de graphes sont utilisés. Le premier consiste en graphes de couches, qui sont nécessaires pour l'allocation de canaux et le routage dans les RCCs semi-statiques. La construction de route dans ce cas de graphe est faite par des algorithmes non optimaux. Le

deuxième type consiste en graphes colorés. Dans ce cas, les solutions utilisées pour découvrir les chemins sont centralisées et heuristiques, ce qui signifie qu'ils peuvent conduire à une route non optimale de routage. Les graphes colorés sont modélisé par $G_c = (N_c, V_c)$ où G_c est le graphe représentant la topologie du réseau, N_c est l'ensemble de sommets et V_c est l'ensemble de arêtes.

Dans le cas où on utilise des méthodes d'optimisation, nous distinguons deux méthodes, qui sont la programmation non linéaire mixte en nombres entiers MINLP (en anglais *Mixed Integer Non-Linear Programming*) et la programmation linéaire mixte en nombres entiers MILP (en anglais *Mixed Integer Linear Programming*). La programmation non-linéaire MINLP introduit une formule qui sert à maximiser le facteur de réutilisation du spectre à travers le réseau. L'objectif est le même pour la programmation linéaire MILP.

Le routage basé sur la connaissance locale du spectre est réalisé de manière distribuée qui prend en considération la gestion locale des ressources radios sur la base d'information partielle sur l'état du réseau. Cette information est recueillie sur la base des paramètres suivants.

- La puissance minimale. Cette information est basée sur le poids minimum du lien entre la source et la destination. Le poids d'un lien est défini selon une fonction basée sur la puissance de transmission. L'emploi des interfaces multiples est considéré. La performance du système proposé est très dépendante de la procédure de découverte de voisinage et la fréquence de mise à jour. Le protocole ne prévoit pas de procédures de maintenance de route rompue suite à l'activité de l'utilisateur primaire (PU).
- Le contrôle d'interférence est fait, soit par le routage au voisin le plus proche NNR (en anglais *Nearest-Neighbor Routing*), où la qualité de service et la distance entre le nœud secondaire et le nœud primaire deviennent des paramètres très importants, soit par le routage au voisin le plus loin FNR (en anglais *Farthest-Neighbor Routing*) du nœud primaire.

- Le délai. La qualité des protocoles de routage est mesurée en termes de délais d'établissement et de maintenance des routes multi-sauts et d'envoi de trafic à travers ces mêmes routes.

- La commutation entre l'accès au medium et la file d'attente. On introduit une métrique considérant à la fois le délai de commutation entre les bandes de fréquences (noté par $D_{\text{switching}}$) et le délai *backoff* (délai d'accès au support) dans une bande de fréquence donnée (D_{backoff}).

- Le Temps de Transmission Effectif (ETT). Il faut minimiser le délai de bout en bout. Les nœuds secondaires interagissent les uns avec les autres afin d'ajuster leurs paramètres de transmission. L'établissement de route sera basé sur le temps affectif ETT considéré comme métrique de routage.

- Construction d'un arbre. Les protocoles basés sur des arbres TBR (en anglais *Tree Based Routing*) ne fonctionnent que sur un seul système sans fil IEEE802.11a ou IEEE802.11b, plutôt que sur plusieurs systèmes sans fil. Les protocoles TBR cognitifs CTBR (en anglais *Cognitive Tree-based Routing*) gèrent plusieurs systèmes. Le protocole STOD-RP (en anglais *Spectrum-Tree based On-Demand Routing Protocol*) est un exemple de ces protocoles (Zhu et al. 2008).

- L'approche probabiliste. Elle est basée sur un calcul probabiliste, tel que pour chaque lien et chaque fréquence, un poids est calculé à partir de la probabilité que la capacité soit supérieure à une demande de communication de débit D (Khalife et Malouch 2009). Le calcul de poids d'un lien est donné par $-\log (Pr[C_k^{(i)} \geq D])$.

Le calcul de la probabilité qu'une capacité $C_k^{(i)}$ soit supérieure à la demande D est:

$$Pr [C_k^{(i)} \geq D] = Pr \left[\frac{P_{ij}^{(i)}}{2^D / W^{(i)} - 1} N_0 \right]$$

La capacité $C_k^{(i)}$ est donnée par $C_k^{(i)} = W^{(i)} \times \log_2 \left[1 + \frac{P_{ij}^{(i)}}{N_0 + P_{ij}^{(i)}} \right]$

où $P_{\eta}^{(j)}$ représente la puissance reçue par le récepteur j , N_0 la densité spectrale de bruit thermique et $W^{(i)}$ la bande passante du canal.

- Le débit. Le but est de maximiser le débit en combinant l'optimisation de délai de bout en bout et la flexibilité des liens, cette flexibilité est due à la disponibilité du spectre à un instant donné et la non-disponibilité à un autre instant.

La solution du problème de routage avec la connaissance locale du spectre prend en considération l'information de disponibilité du canal au voisinage du nœud choisi pour inclure ce nœud dans la route établie selon le protocole de routage.

Dans (Zhai et al. 2010), les auteurs ont proposé une solution nommée FFSR (en anglais *First Fast Second Reliable*). Elle prend en considération la caractéristique dynamique des réseaux ad hoc. Ce protocole s'assure de transférer les paquets de données à travers le plus court chemin dans le but de compléter tout le transfert de données.

Si le lien en cours est invalide, la réparation locale au voisinage de la rupture est déclenchée en premier. Si celle-ci est défectueuse, le nœud source utilise immédiatement le chemin de sauvegarde (*backup*), le temps qu'une nouvelle découverte de route soit faite et le transfert soit complet. Dans cette approche, des nouveautés ont été introduites :

- 1- Un nouveau champ numérique nommé tous-voisins (*all-neighbor*) est ajouté au paquet de contrôle requête de route RREQ (*Route request*). Le paquet de contrôle RREQ est utilisé pour acheminer la demande de route entre une source et une destination. Le champ tous-voisins est utilisé pour compter les nœuds voisins contribuant à construire le chemin.
- 2- L'algorithme ne permet pas aux nœuds intermédiaires de retourner un paquet de contrôle réponse RREP (*Route Response*) s'ils ont l'information du nœud destination dans leur table locale de routage. Le paquet RREP est utilisé pour répondre à une requête de demande de route. Les nœuds doivent continuer de transférer le paquet RREQ le long du chemin jusqu'au nœud destination.
- 3- Le nœud destination peut recevoir plus d'une requête RREQ et donc il calcule pour chaque RREQ le champ (*all-neighbor*). On peut avoir les cas suivants :

- a- La destination reçoit le premier paquet requête RREQ, lit l'information contenu dans ce paquet reçu concernant le nombre de sauts N et le nombre de voisins, et calcule la moyenne. Elle ne retourne pas un RREP qu'après un certain temps.
- b- La destination reçoit un autre RREQ, l'étape (a) est répétée.
- c- La source reçoit les paquets RREPs du nœud destination et le chemin qui compte le moins de sauts est établi entre les deux.
- d- Si un nœud intermédiaire se déplace ou un lien devient invalide, un mécanisme de réparation est déclenché.
- e- Si un chemin est déchu avant la transmission de paquets des données, une réparation locale est lancée.

Un protocole inspiré des protocoles classiques est proposé dans (Cacciapuoti et al. 2010). L'idée de bénéficier de la maturité atteinte par ces protocoles est à l'origine de cette proposition. Le protocole nommé CAODV (en anglais *Cognitive AODV*), qui représente une amélioration du protocole AODV, a été proposé comme solution de routage. Elle est basée sur les principes suivants. En premier il faut éviter les régions d'activité des utilisateurs primaires durant le transfert de paquets pour la formation et la découverte de route sans l'utilisation d'un canal de contrôle dédié. Aussi cette solution doit optimiser un trajet en sélectionnant le canal minimisant le coût d'une route, (ce coût comptera moins de nœuds intermédiaires) ainsi le plus court chemin sera utilisé. Elle doit aussi prendre avantage de la disponibilité des canaux multiples afin d'améliorer les performances globales du réseau.

Le processus de découverte de chemin doit être rapide et résistant aux changements de routes dûs à l'apparition d'activité d'un utilisateur primaire. Chaque utilisateur secondaire est capable d'utiliser les canaux en même temps que les utilisateurs primaires. Cette possibilité est réalisable si l'utilisateur secondaire est équipé de plusieurs interfaces sans fil. Dans le cas d'une seule interface sans fil, la solution suppose la présence d'un mécanisme de coordination de canaux sous-jacents.

De façon similaire à AODV, les RREQs sont diffusés en utilisant une technique de recherche d'expansion du cycle « *route request / route reply* ». A la différence du protocole

AODV, un utilisateur secondaire intermédiaire est supposé recevoir et traiter les RREQs et RREPs entre un sous-ensemble d'un nombre de canaux.

Contrairement à AODV qui considère une route active tant que des données transitent en utilisant cette route, dans CAODV, un nœud peut stocker plusieurs chemins, un pour chaque canal disponible. Aussi les routes peuvent être composées par différents nœuds intermédiaires, qui réfèrent à des chemins différents. Ces chemins restent actifs tant qu'un UP n'est pas signalé. Enfin, le nœud doit vérifier la présence d'un UP avant d'envoyer un paquet à travers une route.

Quand une activité d'un UP est détectée par un nœud sur un canal, le nœud annule toutes les entrées de routage à travers ce canal. Il informe l'utilisateur cognitif voisin que ce canal n'est pas disponible maintenant avec un paquet de PU-RERR. En ce qui concerne la transmission de paquets et afin de maximiser l'efficacité du spectre, le protocole CAODV utilise tous les moyens disponibles pour améliorer les performances globales. L'idée est que la première fois chaque émetteur sélectionne par hasard un des plus courts chemins simples disponibles pour la destination et transmet le paquet de données en utilisant le chemin sélectionné. De cette façon, CAODV bénéficie de la diversité des chemins pour augmenter le taux de livraison de paquets, et également il garde les chemins les plus courts en activité.

Un autre protocole a été proposé dans (Pefkianakis et al. 2008). Il est nommé SAMER (en anglais *Spectrum Aware MESH Routing*). Ce protocole tient compte de la fiabilité des liens en considérant un canal disponible pour long et court terme. Un lien est fiable s'il est utilisé pour transmettre des données sans coupure. SAMER envoie les données sur un canal à long terme sans ignorer les canaux instantanés. Il crée une liste de routes candidates mise à jour périodiquement sur la base d'une métrique mesurant la disponibilité du spectre.

Dans (Kaushik et al. 2010), le protocole CRP (en anglais *Cognitive Routing Protocol for Radio Ad hoc Networks*) est suggéré comme solution de routage. Il permet une protection des récepteurs UP qui sont, généralement, non détectés durant la détection du spectre; l'assignation de plusieurs classes de routes sur la base de différenciation de service de

radios cognitives et une sélection de route faite d'une manière évolutive. Le fonctionnement de ce protocole est basé sur les principes suivants :

- Éviter les zones où il y a des UPs. Les usagers secondaires détectent périodiquement le spectre et voient s'il est disponible.
- Attribuer deux types de classes de routage CR. Le type Classe I attribue une haute importance à la latence de bout en bout, tout en évitant l'interférence avec les UPs. Le type Classe II donne une haute priorité à la protection des UPs en permettant une dégradation de performance des opérations des UCs.
- Sélectionner une route de spectre de façon évolutive. Un canal de contrôle commun est utilisé durant la formation d'une route qui se fait en deux étapes. À la première étape, chaque UC identifie la meilleure bande du spectre sur la base d'observation de l'environnement local et sélectionne le spectre libre. La seconde étape prend en compte une fonction de délai pour l'acheminement du message RREQ, d'où la sélection du prochain saut.

Les métriques étudiées durant l'établissement de route sont données dans la liste suivante:

1. La probabilité de disponibilité de bande passante. Durant l'établissement d'une nouvelle route, le nœud source spécifie la bande passante désirée dans le paquet RREQ.
2. La variance de nombre de bits envoyé sur un lien.
3. L'état de propagation. La caractéristique de propagation de spectre contribue également à réduire la distance par saut permettant ainsi d'atteindre la destination en moins de sauts, ce qui améliore la latence de bout en bout. Lorsque la latence est un facteur, les réseaux RRCs ont tendance à privilégier le spectre avec les meilleures caractéristiques de propagation.
4. La protection des récepteurs UPs. Les régions de chevauchement représentent la zone où les récepteurs UPs peuvent être présents; et donc, il faut minimiser la transmission

de chaque utilisateur cognitif dans le choix de route qui passe par ces UPs, pour les protéger.

5. La considération de détection du spectre. Les utilisateurs CR s'engagent à la détection du spectre à intervalles réguliers pour maintenir à jour l'information concernant l'occupation du spectre. Durant cette durée de détection, et spécialement lorsque la technique de détection d'énergie est utilisée, les utilisateurs CRs adjacents doivent se taire.

La solution ROSA (Ding et al. 2010) (en anglais *Distributed Joint Routing and Dynamique Spectrum Allocation in Cognitive Radio ad hoc Networks*) prend en considération la maximisation du débit qui est un défi majeur dans les réseaux ad hoc à radios cognitives. Sur la base d'un contrôle local, ce protocole vise à maximiser le débit en réalisant conjointement l'allocation dynamique du spectre, l'ordonnancement et le contrôle de puissance de transmission.

La solution proposée permet à chaque utilisateur cognitif de prendre en temps réel une décision sur l'allocation du spectre et la puissance de transmission basée sur des informations collectées localement. Les nœuds peuvent ajuster leur puissance de transmission pour maximiser la capacité de liaison en fonction de la portion des fréquences radios assignées. Un canal de contrôle commun est employé pour la transmission des paquets de contrôle et un autre canal est dédié à la transmission de données. Dans cette solution proposée, le réseau sans fil multi-saut est modélisé comme un graphe de connectivité directe. Le multiplexage OFDM est utilisé dans la couche physique.

Dans (Chowdhury et Felice 2009), les auteurs ont proposé le protocole de routage SEARCH (en anglais *SpEctrum Aware Routing for Cognitive ad hoc networks*) comme solution de routage pour les RRCs. Ce protocole est basé sur le routage géographique qui s'adapte à la disponibilité dynamique du spectre et la mobilité des nœuds, tout en essayant de maintenir la connectivité de bout en bout. La notion géographique de ce protocole signifie l'emplacement des nœuds secondaires par rapport aux nœuds primaires. Il prend en considération l'activité des UPs, en utilisant un routage géographique sur chaque canal pour atteindre la destination avec une identification et un contournement des régions d'activité UP.

Les informations sur le chemin de différents canaux sont ensuite combinées à la destination où le chemin optimal de bout en bout est choisi.

La commutation globale de canal et la décision de détour de chemin sont basées sur la proportion de perte de puissance due à la non-orthogonalité des canaux. Les canaux utilisés par l'utilisateur cognitif peuvent être affectés à différentes zones géographiques dus à la présence d'un seul UP, les zones où se trouve un utilisateur primaire doivent être contournées par les utilisateurs secondaires et aucune connexion entre les utilisateurs secondaires ne peut être établie. Le protocole SEARCH résout ce problème en estimant les détours sur chaque canal en tenant compte de leur couverture. Il calcule ensuite si le coût de la déviation est moindre que le délai de commutation par rapport à la latence de bout en bout.

Le protocole SEARCH utilise un concept de lieux d'ancrage qui se trouvent à être libre de toute activité UP, c'est-à-dire il marque les régions susceptibles d'être occupées par un UP. Dans ce cas, les communications secondaires doivent éviter cette zone. En cas de mobilité des utilisateurs secondaires, l'algorithme vérifie si le prochain saut est à une distance seuil des ancrés. Si cette condition n'est pas satisfaite, le nœud identifie un prochain saut qui est le plus proche du point d'ancrage, garantissant ainsi que les régions UPs soient toujours évitées.

Le protocole SEARCH prédit l'emplacement de la destination à l'avance par l'utilisation de filtre de Kalman de sorte que la route soit convenablement étendue et que les paquets soient livrés de façon fiable au nouvel emplacement. Chaque nœud cognitif a une seule radio et il y a un mécanisme de coordination de canaux sous-jacents à la couche de liaison qui permet aux nœuds voisins à s'engager dans la communication. La maintenance de route est faite selon les cas. Dans le cas de présence d'UPs, les nœuds source et destination doivent changer de position géographique (se déplacer) afin de s'éloigner de UPs et de lancer une nouvelle recherche d'établissement de chemin.

Le protocole PCTC (Guan et al. 2012) (en anglais *Protocol Cognitive Topology Control and routing in Mobile Ad hoc*) est un protocole inter-couche de type «middleware» qui réside entre le module radio cognitif et le routage. À partir de l'idée que les protocoles de routage classiques utilisés pour les réseaux sans fil ad hoc ont atteint une certaine maturité, il est préférable de les utiliser dans les réseaux sans fil ad hoc à radios cognitives. Cette solution

proposée est basée sur la prédiction de la durée du lien et sur la probabilité que cette durée peut vraiment s'étendre jusqu'à la fin de la transmission, tout en considérant à la fois la mobilité des utilisateurs secondaires et les interférences avec les utilisateurs primaires. Le contrôle de topologie est présenté de manière distribuée car il n'est pas pratique de recueillir l'information globale dans un réseau ad hoc mobile à radios cognitives. Ce protocole emploie un poids de lien pour les chemins établis. Il préserve la fiabilité du chemin de bout en bout.

Les auteurs (Sampath et al. 2008) ont proposé le protocole SPEAR (en anglais *SPectrum-Aware Routing protocole*). C'est un protocole de routage pour les réseaux de spectre dynamique. Ce protocole prend en charge la transmission de paquets à haut débit dans le cas d'hétérogénéité du spectre. Le protocole SPEAR est basé sur l'intégration de la découverte du spectre avec la découverte de route pour faire face à l'hétérogénéité du spectre, et pour obtenir une utilisation optimale des canaux disponibles. Il coordonne explicitement l'utilisation de canal sur les nœuds afin d'optimiser l'affectation des canaux sur une base de flux, et de minimiser les interférences inter flux. Et il exploite l'hétérogénéité du spectre local et assigner différents canaux à des liens sur le même flux pour minimiser l'interférence intra flux.

Comme la disponibilité de canaux change avec les activités des UPs, les nœuds effectuent une adaptation locale en modifiant leur utilisation des canaux locaux pour maintenir la connectivité. Si les adaptations locales échouent, SPEAR invoque un nouveau mécanisme de routage pour réparation et restauration de chemin. Les métriques utilisées pour ce protocole sont ETX, ETT et WCETT.

Dans (S. Ju et J. B. Evans 2009), le protocole SARP (en anglais *Spectrum-Aware Routing Protocol for Cognitive Ad hoc Networks*) est constitué de deux fonctions. La première est la fonction de sélection intelligente multi-interface MISF (en anglais *Mutli-Interface Intelligent Selection Function*), et la seconde fonction est la fonction de sélection multi-chemin MPSF (en anglais *Multi-Path Selection Function*).

Le protocole fonctionne comme suit. Si chaque nœud a de multiples interfaces alors chaque interface écoute différents canaux, ce qui revient à dire qu'un nœud peut utiliser une ou deux interfaces pour acheminer ses paquets. Ainsi les nœuds peuvent éliminer ou réduire l'interférence entre nœuds voisins. La charge du réseau est toujours balancée.

La première fonction MISF utilise comme métrique le délai des paquets RREQ, tandis que MPSF utilise l'augmentation de bande passante comme métrique. Aussi, le protocole SARP emploie une base de données historique pour enregistrer les informations des couches MAC et physique. Il a besoin de ces informations pour connaître le type de canaux et leurs capacités.

2.3 Synthèse de l'état de l'art

L'étude et la synthèse de la littérature citée précédemment nous permet de conclure avec les commentaires suivants.

Dans (Cesana et al. 2010 ; Zhai et al. 2010; Cacciapuoti et al. 2010), nous remarquons une similitude entre les solutions dû au fait qu'elles se basent sur les principes des protocoles classiques, avec quelques ajouts dans le but d'améliorer et d'adapter au mieux l'algorithme de routage aux RRCs.

La première solution a ajouté l'emploi de deux interfaces, l'une fixe et l'autre commutée, avec ajout d'une liste de voisins dite Table Voisins. La seconde solution est basée sur une découverte rapide de chemins sur la base d'un calcul de la moyenne de nombre de sauts par rapport au nombre de voisins. La troisième solution permet l'utilisation de canal en absence de UPs. Dans le cas contraire, tout le voisinage du nœud où l'activité UP est signalée ne devient plus utilisable, il est considéré comme zone marquée.

Nous pouvons déduire que le protocole de routage à utiliser dans les réseaux à radios cognitives doit profiter de la maturité des protocoles de routage classiques en les adaptant aux réseaux à radios cognitives. Il doit aussi se baser sur l'emploi des équipements sans fil multi-interfaces cognitifs et sur l'emploi d'une base de données afin de récolter les informations de l'environnement avoisinant.

Le reste des propositions nous a poussé à regarder de près les recherches faites sur les réseaux RRCs et principalement sur les protocoles inter-couches MAC et physique. Nous savons que les nœuds cognitifs peuvent compter sur eux-mêmes pour former un réseau

hétérogène vis-à-vis des réseaux primaires. Leur façon de communiquer dépendra étroitement de la performance du protocole de routage.

Il est vrai que les RRCs présentent une ressemblance avec les réseaux multi-radios multi-canaux maillés, mais la technologie radio cognitive ajoute de nouveaux défis tels que le traitement des transmissions sur des canaux parallèles et la réduction des interférences entre les utilisateurs primaires et les utilisateurs secondaires. Les utilisateurs secondaires possèdent des capacités physiques qui permettent de détecter, de commuter, et de transmettre sur de nombreuses bandes de spectre. Si la bande de fréquences allouée au primaire reste inutilisée pendant une durée indéterminée et non-limitée, alors l'environnement considéré est dit statique ce qui permet d'utiliser les protocoles de routages classiques et de bénéficier de leur maturité. Si l'arrivée et le départ des transmissions des nœuds primaires sont plus dynamiques alors des nouvelles approches cognitives spécifiques à cet environnement doivent être développées.

Le protocole doit prioritairement trouver un chemin disponible et stable. La stabilité de chemin peut être assurée en incluant l'information de sélection du spectre du chemin dans l'algorithme, ou par des métriques de routage qui captent les fluctuations du spectre et favorisent les bandes stables. Le calcul doit être rapide et permet les changements dynamiques (L. Gavrilovska et V. Atanasovski 2012).

En analysant ce qui a été dit précédemment, nous pouvons en tirer les idées suivantes.

- Le routage dans les réseaux RRCs pourrait envisager de multiples canaux de transmissions parallèles, alors que la sélection du canal et la gestion du spectre ne doivent être traités qu'à la couche MAC. L'interaction entre la couche MAC et la couche réseau doit être soigneusement étudiée.
- La sélection du canal par la couche MAC cognitive implique la sélection du prochain saut qui conduira à différents voisins et pourrait être optimisée localement en se basant sur les informations MAC relatives pour choisir les prochains sauts.

- Les chemins obtenus peuvent ne pas être optimaux pour tous les flux dans le réseau RRC car la sélection ne tient compte que des informations locales et n'a pas de vision globale du réseau.
- L'exécution de routage à la couche réseau tout en utilisant une fonctionnalité de la couche MAC comme métrique de routage serait une source d'instabilité même si les routes construites sont optimales en considérant l'interférence et la puissance de transmission.
- L'amélioration de routage avec l'information de la couche MAC est une technique d'inter-couches qui a un effet non négligeable sur la charge (*overhead*) et la complexité de l'algorithme de routage. Une solution entre ces deux approches nécessite de trouver un compromis entre la faisabilité et la stabilité d'un côté et la réactivité et la complexité de l'autre.
- Un bon compromis est d'aider le routage par une métrique de couches inférieures qui restent utilisable pour une longue période de temps. Cette métrique devrait également refléter la disponibilité et la qualité du spectre.

Toutes les idées citées ci-dessus peuvent nous servir dans la conception de notre solution qui est à la base le protocole de routage réactif TORA avec des modifications pour l'adapter aux réseaux RRCs.

CHAPITRE III

CONTRIBUTION

3.1 Présentation du protocole de routage réactif TORA

Le protocole de routage avec classement temporaire TORA (en anglais *Temporary Ordering Routing Algorithm*) a été conçu par V. D. Park et M. S. Corson (V. D. Park et M. S. Corson 1997, 2). C'est un protocole de routage distribué où chaque nœud contient une copie de l'algorithme afin de l'exécuter localement. Comme indiqué dans son nom, le protocole est basé sur une notion de temps dans le sens où les nœuds ont une horloge synchronisée utilisée durant l'exécution. L'algorithme TORA utilise une technique de routage appelée "inversement de liens" (en anglais *Link Reversal*). Cette technique permet, en cas de défaillance d'un lien entre la source et la destination, au protocole de réagir rapidement en inversant le lien entre ces deux nœuds et en établissant un nouveau chemin. L'algorithme TORA permet de créer des routes, de les maintenir, d'éliminer les routes rompues et même d'optimiser les routes établies.

L'algorithme TORA est appliqué sur les graphes acycliques orientés DAG (en anglais *Directed Acyclic Graph*) modélisant le réseau. Il essaye de maintenir une propriété appelée orientation destination des graphes orientés. Le graphe devient non orienté si un lien devient défaillant. Le graphe acyclique orienté emploie une notion de hauteur de nœud de sorte que chaque nœud communique sa hauteur à tous ses voisins directs afin que l'orientation de lien entre deux nœuds voisins se fasse du nœud ayant la plus grande hauteur vers celui ayant la plus petite (fig. 3.1). Cette hauteur représente la métrique de routage du protocole TORA.

Elle correspond au quintuple défini comme suit :

- l'instant de rupture de lien,
- l'identificateur du nœud qui définit le nouveau niveau de référence,
- le bit indicateur de réflexion permettant d'indiquer un nouveau niveau,
- un paramètre d'ordonnancement de propagation, et finalement
- l'identifiant unique du nœud.

La valeur "zéro" désigne la plus petite valeur d'une hauteur. La valeur "Nulle" désigne une valeur indéfinie d'une hauteur.

Le protocole TORA a pour but de minimiser l'effet de mobilité sur le routage, qui est fréquente dans les réseaux ad hoc et qui est la première cause de rupture de route. Pour le faire, il stocke plusieurs chemins vers une même destination. L'algorithme de routage TORA maintient les routes au niveau local (voisinage) de la défaillance du chemin c.-à-d. au niveau des nœuds les plus proches de la route rompue. Les messages de contrôle employés sont réduits et ne sont destinés qu'aux nœuds voisins localisés près de la rupture du chemin, donc à une partie restreinte du réseau, ce qui limite la charge du réseau.

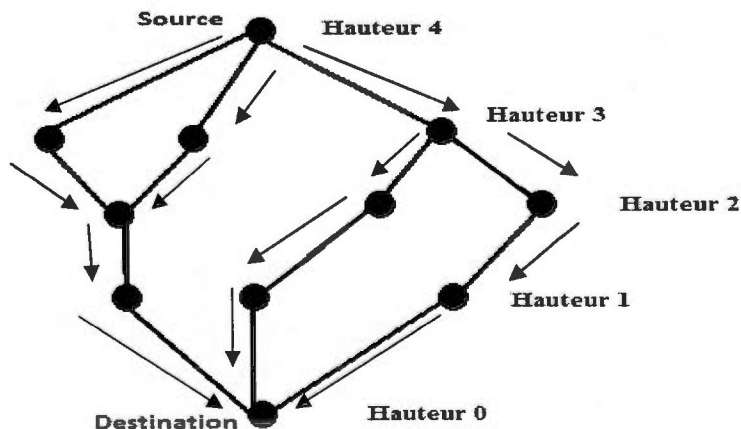


Figure 3.1 Modélisation de chemins par graphe acyclique orienté dans le protocole de routage TORA (V. D. Park et M. S. Corson 1997, 2)

Le fonctionnement du protocole TORA est basé sur quatre fonctions : création de route, maintenance de route, effacement de route et optimisation de route.

- Création des routes

Le fonctionnement traditionnel de l'algorithme TORA pour créer une route se déroule comme suit. Le nœud source diffuse un paquet requête nommé demande d'établissement de route QRY (*QueRY*) en indiquant l'identifiant du nœud destination. Il envoie la requête à ses voisins directs qui, à leur tour, le transmettent à leurs voisins jusqu'à la destination. Le destinataire du paquet QRY répond par l'envoi d'un paquet réponse UPD (*UPDate*). Au début du processus tous les nœuds du réseau ont la hauteur indéfinie "Nulle" sauf la destination qui a la hauteur "Zéro". Le changement des valeurs de hauteur d'un nœud est fait par le biais des paquets de contrôle QRY et UPD (V. D. Park et M. S. Corson 1997, 2).

Si un paquet QRY est reçu par un nœud alors :

- a) Si ce nœud n'a aucun lien sortant et son drapeau (indiquant si une route est requise) n'est pas marqué, alors il marque son drapeau et diffuse le paquet QRY à ses voisins directs.
- b) Si ce nœud n'a aucun lien sortant et son drapeau est marqué, alors il rejette le paquet QRY.
- c) Si ce nœud a un lien sortant et sa hauteur est "Nulle" alors il incrémente de 1 sa hauteur (sa hauteur est égale à la hauteur minimale de ses voisins non "Nulle") et envoie un paquet UPD à ses voisins directs.

Si un paquet UPD est reçu par un nœud alors :

- a) Si le drapeau de ce nœud est marqué alors le nœud incrémente sa hauteur de 1, supprime la marque de son drapeau et diffuse un paquet UPD à ses voisins directs.
- b) Si le drapeau de ce nœud n'est pas marqué alors il met à jour simplement ses entrées relatives aux états de lien.

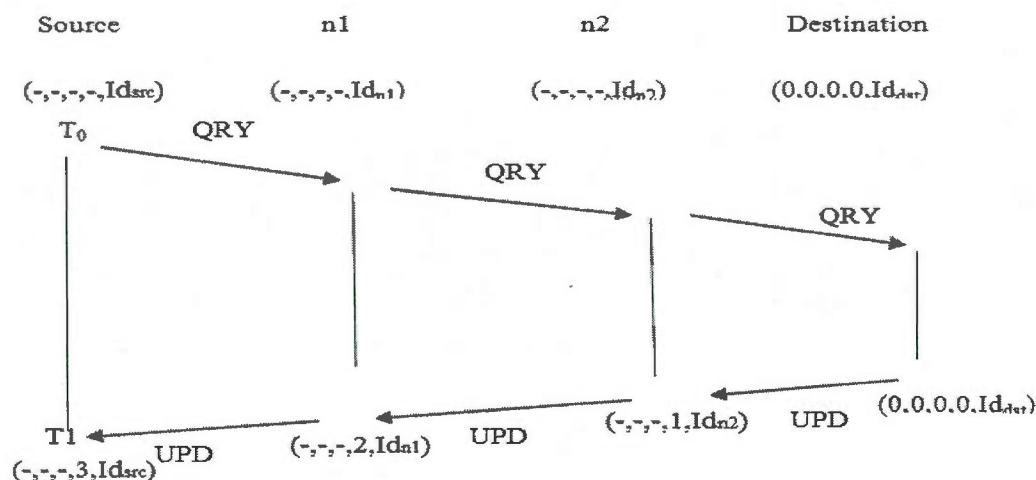


Figure 3.2 Processus de création de route (V. D. Park et M. S. Corson 1997, 2)

Dans la figure 3.2 à l'instant T_0 la source fait une demande d'établissement de communication par l'envoi d'un paquet QRY à son voisin direct le nœud n_1 . Ce paquet contient l'information de la destination ainsi que la hauteur du nœud (Nulle). Le nœud n_1 à son tour diffuse le paquet QRY à ses voisins directs (dans le scénario de la figure 3.2 c'est le nœud n_2), il transmet sa valeur hauteur égale à "Nulle" et son identifiant. Le processus de diffusion du paquet QRY se poursuit jusqu'à la destination qui contient une hauteur égale à "Zéro". À l'instant T_1 le nœud destination répond à cette requête par l'émission d'un paquet UPD et transmet sa hauteur égale à "Zéro" au nœud expéditeur du paquet QRY. À l'arrivée du paquet UPD au nœud n_2 , le nœud modifie sa hauteur en ajoutant 1 à la hauteur contenue dans ce paquet et il envoie un paquet réponse avec sa nouvelle hauteur au nœud n_1 . Le même processus se répète au niveau du nœud n_1 qui transmet un paquet UPD avec sa nouvelle hauteur au nœud source. Le nœud source reçoit le paquet UPD, modifie sa hauteur et établit la connexion vers la destination. Ainsi, le chemin est construit de la plus haute hauteur (dans la figure 2.3 la valeur de la hauteur du nœud source est égale à 3) vers la plus petite hauteur (la valeur de la hauteur du nœud destination est égale à 0).

- Maintenance des routes

Suite à une défaillance d'un lien, une réparation de route doit être faite. Un paquet UPD est envoyé à tous les nœuds voisins par le nœud où la rupture est détectée. La valeur de la métrique hauteur de ce nœud est non "Nulle" (V. D. Park et M. S. Corson 1997, 2).

Le processus de maintenance de route est moins coûteux en paquets de contrôle que l'établissement d'un nouveau chemin car l'algorithme analyse l'état des nœuds les plus proches de l'endroit de la rupture. Cela se fait selon les deux cas suivants. Dans le premier cas, le nœud traité n'a aucun lien sortant après un échec de lien. Donc, il doit définir un nouveau niveau de référence et mettre à jour la valeur correspondante au niveau du quintuple de chaque nœud. Dans le deuxième cas, le nœud traité n'a aucun lien sortant car une inversion de lien a été faite après la réception d'un paquet UPD. Dans ce cas, il faut vérifier si :

- Les 3-tuples (instant de rupture de lien, identificateur du nœud créateur de niveau de référence, bit indicateur de réflexion) pour tous les nœuds voisins ne sont pas égaux alors le nœud propage le plus grand niveau de référence de ses voisins et choisit la plus petite hauteur correspondante à ce niveau de référence.
- Les 3-tuples (instant de rupture de lien, identificateur du nœud créateur de niveau de référence, bit indicateur de réflexion) pour tous les nœuds voisins ne sont pas égaux et le bit indicateur de réflexion égal à "zéro", alors le nœud propage le même niveau de référence à ses voisins.
- Les 3-tuples (instant de rupture de lien, identificateur du nœud créateur de niveau de référence, bit indicateur de réflexion) pour tous les nœuds voisins ne sont pas égaux et le bit indicateur de réflexion est égal à "un", alors le nœud doit lancer le processus d'effacement de routes.

- Effacement des routes

Cette fonction utilise le paquet de contrôle CLR (*CLear*). Au niveau du nœud à traiter, il faut mettre la valeur de la hauteur à "Null" ainsi que celles de tous ses voisins directs.

- Optimisation des routes

Cette fonction utilise des paquets de contrôle OPT (*OPTimized*). Elle est employée lorsque tous les chemins vers une destination sont découverts. Il est à signaler que tous les paquets transmis par TORA d'un nœud source vers un nœud destination doivent être encapsulés par le protocole IMEP (en anglais *Internet MANET Encapsulation Protocol*) défini par (V. D. Park et M. S. Corson 1999), avant d'être transmis. IMEP fait la liaison de ces paquets à la couche liaison. Le protocole TORA est informé par le protocole IMEP de tout changement d'état des liens vers les voisins via des paquets de contrôles (*beacons*).

3.2 Caractéristiques et avantages

Le protocole de routage TORA présente les caractéristiques et avantages suivants. Il réagit rapidement au changement dans la topologie du réseau dû à la mobilité. Il diminue la charge du réseau « *overhead* » dû au maintien local des ruptures des liens. Pour le protocole TORA, toutes les routes sont sans boucle, ce qui assure un transfert de données en temps limité. Aussi, chaque destination peut avoir plus d'une route. TORA est un protocole multi-chemins. Pour TORA, chaque nœud possède l'information de tous ses voisins directs et peut transmettre des paquets traversant plusieurs autres nœuds voisins jusqu'à la destination, ce qui donne l'aspect multi saut au protocole TORA.

3.3 Hypothèses associées à notre proposition

Afin de développer une solution au problème de routage dans les réseaux RRCs ad hoc, il faut prendre en considération les idées tirées de l'analyse faite sur la littérature pertinente, faire un rapprochement avec le protocole de routage TORA, puis développer les idées à ajouter pour que ce protocole réponde au mieux aux exigences des réseaux RRCs, enfin, adapter ce protocole à un environnement de RRCs. L'évaluation de cette solution se fera par une étude des performances par simulation en utilisant le simulateur NS-2, ce qui sera notre prochaine étape.

Nous considérons pour notre solution un environnement opportuniste. Il faut donc que la solution soit capable de trouver une route entre une source et une destination dans un

temps réduit afin de transférer les données avant l'apparition d'un nœud primaire. En se basant sur la description du protocole de routage TORA, nous pouvons dire que cela est possible vu que TORA construit un chemin source-destination par des requêtes envoyées d'un nœud à tous ses voisins directs. La création du chemin est plus rapide si TORA transmet ces paquets de construction de route plus rapidement sur plusieurs canaux ce qui nous mène vers la proposition d'une solution supportant le concept multi-interface multi-canal. Il faut toujours se rappeler que TORA est un protocole de routage multi-saut.

Nous supposons aussi que l'apparition d'un nœud primaire engendrera une cessation automatique du trafic et un relancement du processus de découverte de nouvelles routes. TORA permet cela grâce à la construction locale d'une route rompue. Le processus de maintenance de route au niveau local peut être employé.

La distinction entre le trafic des nœuds primaires et des nœuds secondaires doit être ajoutée à notre approche. Elle nous permettra de rompre le trafic entre des nœuds secondaires en présence de transmission des nœuds primaires. Nous devons modifier et adapter TORA en introduisant une différenciation entre les deux types de paquets transmis.

3.4 Description de la solution proposée

Suite aux hypothèses posées précédemment, nous proposons une solution qui bénéficie de la maturité du protocole de routage TORA. Nous portons les modifications nécessaires afin que ce protocole supporte le multi-interface, ce qui permettra d'utiliser plusieurs canaux de transmission et donnera la possibilité de transmettre les données plus rapidement et d'éviter la rupture de chemin et l'arrêt de transmission, car si une transmission sur un canal donné n'est plus possible elle le sera sur un autre. Cette modification ne suffit pas pour adapter TORA aux RRCs. Il faut aussi modifier le protocole afin qu'il puisse supporter les réactions nécessaires à la présence d'activité de nœuds primaires. Nous proposons de distinguer les paquets de contrôle employés pour l'établissement de chemin par les usagers primaires et les usagers secondaires. Dans le cas où une réception d'un paquet de contrôle primaire est détectée, un processus de cessation d'activité des nœuds secondaires qui ont reçu ce paquet doit être déclenché. La cessation est déclenchée par l'envoi d'un paquet de contrôle CLR afin de rompre tous les liens sortant et entrant vers ce nœud. En parallèle, un processus de maintenance de route est déclenché chez les nœuds voisins du chemin rompu.

CHAPITRE IV

MISE EN ŒUVRE DE LA SOLUTION PROPOSÉE ET SIMULATION

4.1 Présentation de l'outil de simulation NS

NS (en anglais *Network Simulator*) est un simulateur à événements discrets. Les chercheurs le considèrent comme un des meilleurs moyens de simulations des réseaux, car il permet d'étudier une grande partie des fonctionnalités des couches d'un réseau, que se soit au niveau physique, au niveau liaison, au niveau réseau et même au niveau application.

Il permet de simuler des réseaux filaires ou des réseaux sans fil. NS comporte les fonctions des protocoles de routages DSDV, AODV, DSR (en anglais, *Dynamique source routing*) et TORA, des protocoles de la couche liaison au niveau MAC tel que 802.11, et des protocoles et applications de la couches application tel que le transfert de fichier FTP (en anglais, *File Transfert Protocol*) qui fonctionne sur TCP (en anglais, transport control protocol) et le trafic CBR (en anglais *Constant bit rate*) qui fonctionne sur UDP (en anglais, *user datagram protocol*) (The network simulator NS-2, 2012).

De plus, c'est un logiciel libre (*open source*) ce qui facilite la modification et l'adaptation de ses fonctions au besoin des recherches. Il est écrit en langages de programmation C++ et OTCL (*Orienté objet du langage TCL*). Il est à signaler que malgré que NS supporte plusieurs protocoles de routage multi-sauts, il ne supporte pas les protocoles de routage multi-interfaces.

4.2 Mise en œuvre de la solution

4.2.1 Modification du protocole TORA au concept multi-interface

Pour apporter les modifications associées au modèle multi-interface à notre solution, nous avons étudié deux solutions qui ont traités l'adaptation de protocole classique par exemple le protocole AODV au modèle multi-interface.

La solution Hyacinth (R. A. Calvo et J. P. Campo, 2007) a été faite dans le but d'introduire le multi-interface dans NS-2.1b9a. Elle permet d'affecter de façon statique un certain nombre d'interfaces à un nœud donné. Cependant, cette technique montre des limites en cas d'ajout ou de réduction du nombre d'interfaces. De plus, il n'est pas précisé comment introduire ces modifications pour que le protocole de routage les supporte.

Une autre solution proposée par R. A. Calvo et J. P. Campo (R. A. Calvo et J. P. Campo, 2007) a comblé les lacunes de la solution citée précédemment. Un modèle multi-interface multi-canal est adapté pour le protocole de routage AODV, où l'architecture des nœuds sans fil a été modifiée afin de permettre l'implémentation de protocole de routage multi-interface multi-canal. Cette modification nous permettra d'implémenter notre solution multi-interface sur NS-2 et de pouvoir la simuler. Dans NS-2, un nœud sans fil a une architecture comme montré dans la figure 4.1.

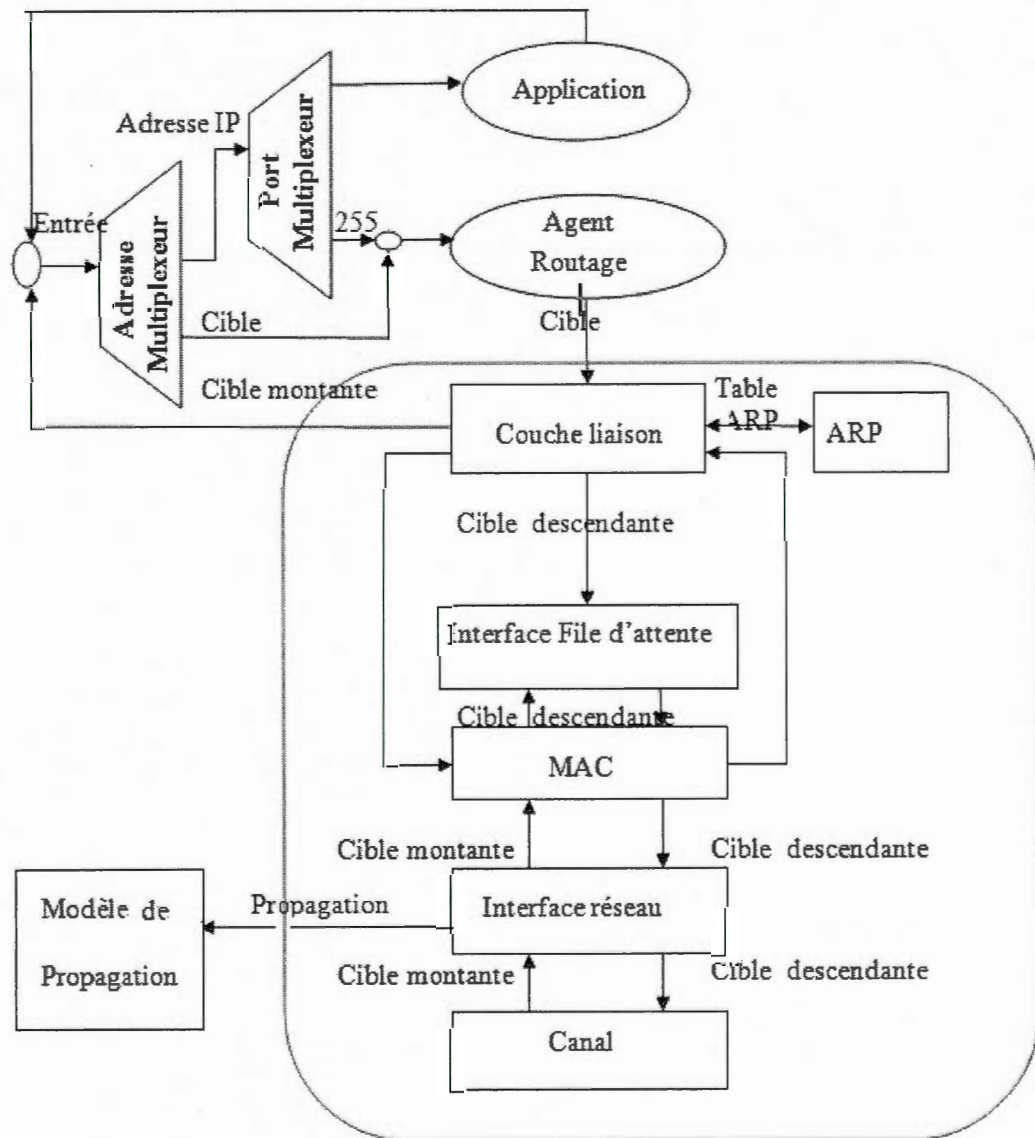


Figure 4.1 Architecture d'un nœud mobile dans NS-2 (R. A. Calvo et J. P. Campo, 2007)

L'adaptation d'un nœud multi-interface consiste à apporter des modifications au niveau de la couche liaison, la table ARP (en anglais, *address resolution protocol*), la file d'attente, la couche MAC, l'interface réseau et le canal. Ces modules sont encadrés en rouge dans la figure 4.1. Chaque module sera présenté par une liste où chaque élément de cette liste

contient la même présentation (fig. 4.2) ce qui transforme l'architecture précédente comme suit.

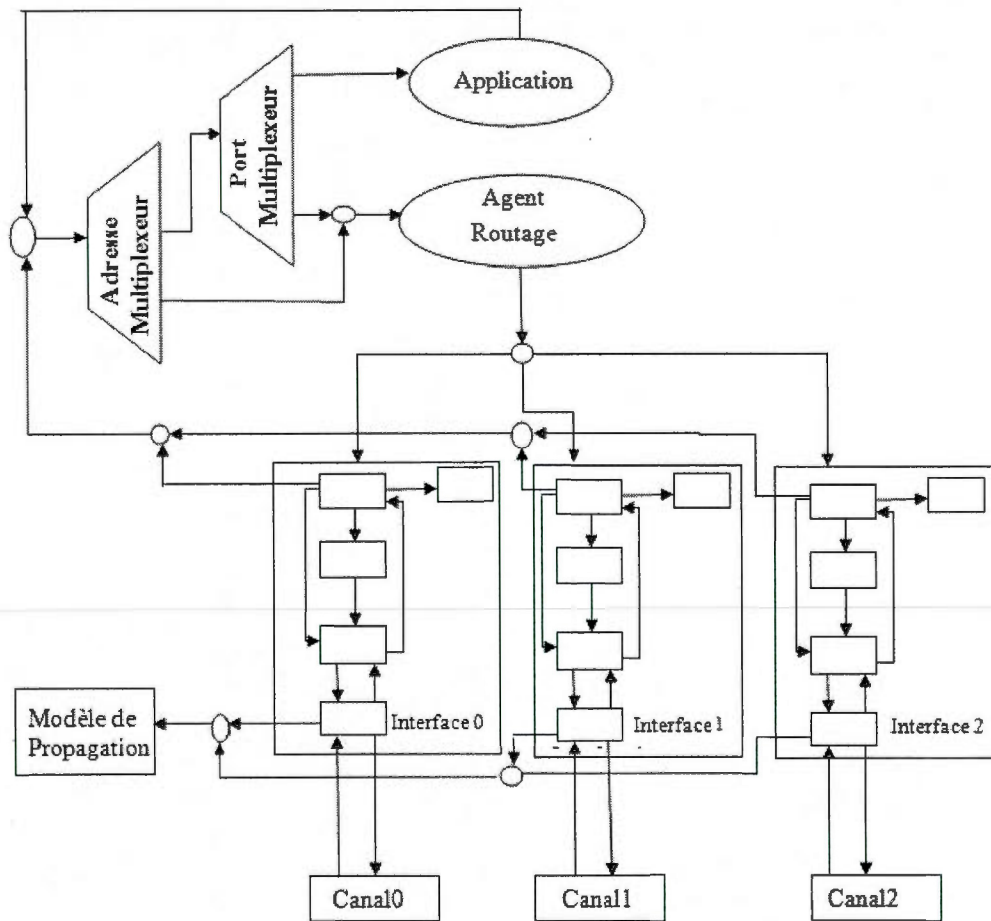


Figure 4.2 Architecture d'un nœud mobile multi-interface adaptée par (R. A. Calvo et J. P. Campo, 2007 ; P. Lee et G. Wei 2009)

L'implémentation de cette architecture dans NS-2 se fait en trois étapes principales. Une première étape sera consacrée à modifier les procédures au niveau des fichiers TCL dans NS-2. Dans la deuxième et troisième étape nous introduisons les modifications au niveau des fichiers .cc et .h.

Première étape :

La première étape consiste à introduire de nouvelles procédures afin de gérer les interfaces et d'ajouter des canaux. Afin de réaliser cela, nous nous basons sur le travail (R. A. Calvo et J. P. Campo, 2007). Nous programmons les procédures qui permettent la gestion de plusieurs interfaces aux nœuds de NS-2 et ce qui donne la possibilité d'utiliser un protocole de routage multi-interface. Ces procédures sont :

- "change-numifs" permet de changer le nombre d'interfaces utilisés dans la communication où la variable numifs représente le nombre d'interfaces,
- "get-numifs" permet de savoir combien d'interfaces sont attribuées à un nœud donné,
- "add-channel" permet d'affecter un canal à une interface et
- "if-num" permet de tester si une ou plusieurs interfaces ont été utilisées.

Toutes ces procédures doivent être programmées au niveau du fichier ns-lib.tcl. En suite, il faut ajouter le multi-interface au niveau des procédures de configuration des nœuds node-config et create-wireless-node dans le fichier ns-lib.tcl.

Deuxième étape :

La deuxième étape consiste à ajouter la gestion de table ARP par interface dans la procédure add-interface, à modifier la procédure add-target pour déterminer le nombre d'interfaces utilisé par l'appel de la procédure get-numifs. (Pour la procédure add-target-rtagent nous devons programmer la partie concernant le protocole de notre étude TORA, en prenant en considération la partie où le traitement IMEP est fait dans le fichier ns-mobilenode.tcl) et à mettre à jour les procédures "init-args" et "reset" pour supporter plusieurs tables ARP (Chaque interface a sa propre table ARP),

Troisième étape :

Cette étape concerne le changement au niveau des procédures dans les fichiers « mobilenode.h », « mobilenode.cc », « channel.cc » et « mac-802_11.cc ». Elle consiste à introduire la gestion d'indice pour chaque interface car nous avons utilisé des tables dans notre conception comme cité dans l'étape précédente. Chaque indice d'une table correspond à une interface. En suite il faut modifier la procédure recv() au niveau MAC afin de récupérer l'information de toutes les interfaces définies pour un nœud donné. Il faut ajouté dans la procédure recv() une itération de tout les interfaces définies pour récupérer tout les paquets

reçus. Il est très important de comprendre le fonctionnement de TORA et les échanges avec le protocole IMEP. Il lui est directement relié comme une sous-couche tel qu'indiqué dans la figure 4.3.

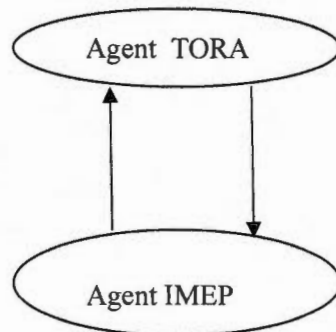


Figure 4.3 Interaction du protocole TORA avec le protocole IMEP (V. D. Park et M. S. Corson 1999)

Il est nécessaire d'apporter les modifications au protocole de routage TORA pour introduire le concept de multi-interface. Dans notre cas, le protocole TORA utilise un autre protocole appelé IMEP (protocole d'encapsulation de MANET pour Internet), qui sert à vérifier l'état des liens et à encapsuler les paquets de contrôle de TORA pour les transmettre au niveau liaison. Donc les modifications doivent aussi toucher le protocole IMEP.

Toutes les procédures qui sont à l'origine d'envoi ou de réception de paquets sont modifiées en introduisant le nombre d'interfaces dans l'envoi de paquets. Ces procédures se trouvent dans les fichiers «tora.h», «tora.cc», «tora-io.cc», «imep.h» et «imep.cc». Une fois que ces modifications accomplies, nous pouvons lancer des simulations pour évaluer la performance de notre nouveau protocole. Il faut signaler que nous avons choisi une version stable de NS-2, la version 2.30, afin d'éviter les bugs présents dans les nouvelles versions.

4.2.2 Adaptation du protocole TORA aux RRCs

4.2.2.1 Format des paquets de routage employés par TORA

TORA utilise trois types de paquets dans l'établissement ou la destruction d'un chemin, le format et le rôle de chaque paquet est défini par la suite (V. D. Park et M. S. Corson 1997, 2) :

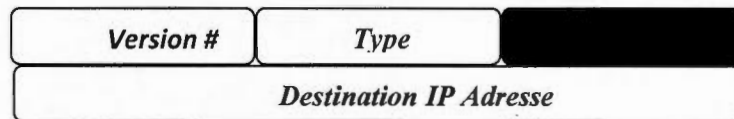


Figure 4.4 Format d'un paquet QRY (*QueRY*) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)

Dans le cas de paquet QRY (fig. 4.4), le champ Type est 1. Ce paquet est envoyé au début de la découverte de chemin à tous les voisins du nœud source. Le champ nommé "Réserve" dans ce paquet est non utilisé, ce qui nous permettra de l'exploiter dans notre solution et en y insérant l'information 'Nœud Primaire' ou 'Nœud Secondaire'. Ainsi à l'arrivée de ce paquet à un nœud intermédiaire ou au nœud destination, une vérification pourra être faite sur ce champ afin de déterminer l'action suivante à entreprendre dès sa réception.

Si le paquet provient d'un nœud primaire et le nœud qui le reçoit est un nœud secondaire, un processus de rupture de chemin est lancé à partir de ce nœud secondaire. Le processus de maintenance de route est déclenché pour une réparation rapide de chemin et une nouvelle connexion doit être trouvée.

L'établissement d'une nouvelle route ne se fait pas seulement avec les paquets QRY. Des paquets UPD sont utilisés pour acheminer la requête de découverte de chemin. Ce paquet a le format indiqué dans la figure 4.5.

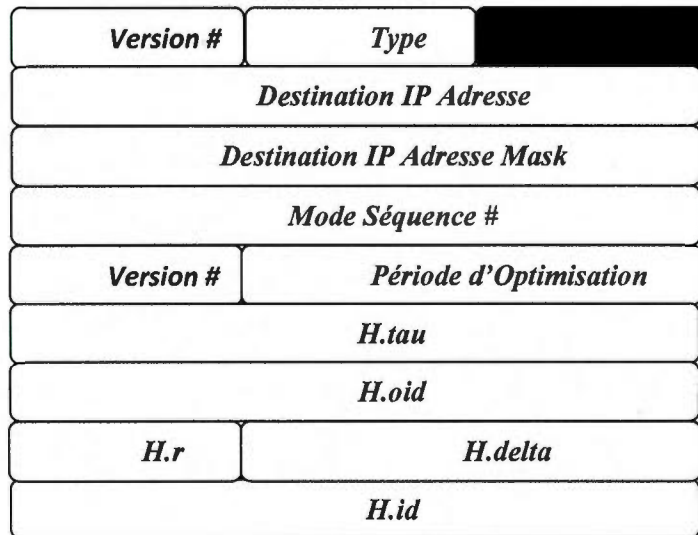


Figure 4.5 Format d'un paquet UPD (*UPDate*) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)

Dans ce paquet le champ Type est 2. Le champ "Réserve" est non utilisé, ce qui nous servira pour distinguer le nœud primaire du nœud secondaire. Sur la base de la valeur contenue dans ce champ, le protocole de routage fonctionnera. Si les paquets proviennent d'un nœud primaire alors tous les nœuds secondaires qui reçoivent ce paquet rompent leurs liens vers leurs voisins directs et un nouveau processus de maintenance de chemin se déclenchera au niveau local de ces nœuds (au voisinage des nœuds qui ont rompu leurs liens).

Le paquet de routage utilisé par TORA pour rompre une liaison entre un nœud et ses voisins est le paquet CLR. Le format de CLR est comme dans la figure 4.6.

<i>Version #</i>	<i>Type</i>	
<i>Destination IP Adresse</i>		
<i>Destination IP Adresse Mask</i>		
<i>H.tau</i>		
<i>H.oid</i>		
<i>H.id</i>		

Figure 4.6 Format d'un paquet CLR (*CLEAR*) du protocole TORA (V. D. Park et M. S. Corson 1997, 2)

Le champ Type dans un paquet CLR a la valeur 3. Le champ "Réservé" ne contient aucune valeur préalable ce qui est en notre faveur pour l'employer comme champ distinguant le type de nœud qui a transmis le paquet (nœud primaire ou nœud secondaire).

En conclusion, tous les paquets de routage employés par le protocole TORA contiennent un champ réservé que nous pouvons utiliser pour acheminer l'information qui distingue les nœuds expéditeurs du paquet (les nœuds secondaires de ceux primaires). Cette information est importante pour donner la priorité aux paquets originaires de nœuds primaires par rapport au trafic des nœuds secondaires. Ces derniers doivent cesser toute activité ou chercher d'autres routes pour continuer leur communication.

La programmation à faire dans ce cas concerne essentiellement les fichiers "tora-packet.h", "tora-io.cc" et "tora.cc". Les modifications que nous introduisons définissent un test de vérification des paquets reçus pour identifier si le paquet provient d'un nœud primaire ou d'un nœud secondaire et ainsi faire le traitement adéquat. Les procédures sendQRY, sendUPD et sendCLR servant à transmettre les paquets de contrôle QRY, UPD et CLR. Les modifications sont faites par l'introduction d'une affectation de la valeur 1 au champ "Réservé" dans le cas où l'expéditeur est un nœud secondaire. Pour les procédures recvQRY, recvUPD qui sont destinées à la réception des paquets de contrôle QRY ou UPD, elles sont modifiées en ajoutant un test au début des procédures. Si à la réception de paquet QRY ou de paquet UPD la valeur du champ "Réservé" est différente de 1 alors une rupture de connexion par l'envoi d'un paquet CLR est faite au niveau des nœuds secondaires qui ont reçu ce

paquet. Un processus de maintenance de route est lancé pour trouver d'autres chemins et rétablir la communication rompue.

4.3 Simulation et évaluation des performances

4.3.1 Paramètres de simulation

Dans cette section nous définissons les différents paramètres nécessaires pour notre simulation. Nous utilisons un réseau à 10 nœuds sans fil mobiles. Ces nœuds sont positionnés initialement de façon aléatoire par le simulateur de mobilité dans NS-2. Ces positions sont dans une zone carrée de 670×670 m qui est la topologie de simulation. Après un intervalle de temps défini, les nœuds commencent leur mouvement selon un modèle de mobilité *random way point* et une vitesse de déplacement variant entre zéro et une vitesse maximale fournie comme paramètre de simulation. Chaque nœud stoppe son mouvement durant un temps de pause (dit *pause time*) avant le prochain saut qui le conduira vers sa prochaine position définie de manière aléatoire.

Afin de simuler un environnement opportuniste où les nœuds ont une très forte mobilité, nous avons utilisé des valeurs de temps de pause de 0s, 10s, 30s, 50s et 100s et nous avons fixé la vitesse de mobilité à 20 m/s pour étudier l'effet du changement fréquent des nœuds sur notre nouveau protocole multi-interface. Dans un second temps, nous avons utilisé des vitesses de mobilité de 20 m/s, 40 m/s, 60 m/s, 80 m/s et 100 m/s et nous avons fixé le temps de pause à la valeur 0s. Donc, les nœuds sont toujours en mouvement et en changement de chemin. Ces différents scénarios sont exécutés durant 150 secondes qui est la durée de simulation. Les simulations sont exécutées dans un réseau à multi-sauts à base de IEEE 802.11. Le modèle de propagation utilisé est *Two-ray-ground*. C'est un modèle fourni avec NS-2. Il est utilisé pour prédire la puissance avec laquelle chaque paquet est reçu. Il est destiné pour les grandes distances ($1/r^4$). De plus, un module d'antenne omnidirectionnel (*Omni Directionan Antenna Module*) est utilisé. Au niveau de la couche liaison les paquets sont fragmentés et réassemblés. Les adresses IP sont aussi converties en adresses MAC. Pour la file d'attente utilisée dans notre simulation nous avons considéré le paramètre *Queue / DropTail / PriQueue* qui donne la priorité au paquet de contrôle ce qui répond le mieux à

notre étude de protocole de routage. Cette file d'attente peut contenir jusqu'à 50 paquets. Notre simulation est faite sur un réseau sans fil ce qui nous oblige à configurer au niveau de la couche physique l'interface à la valeur *Phy / WirelessPhy* qui correspond à une interface sans fil. Pour les canaux, nous utilisons la valeur correspondante aux canaux sans fil *Channel / WirelessChannel*. Ces paramètres sont résumés dans le tableau 4.1.

Paramètres	Valeurs
Dimension de la zone	670 × 670 m
Nombre de nœuds	10
Temps de simulation	150 s
Couche Mac	IEEE 802.11
Type de propagation	Propagation/TwoRayGround
Type d'antenne	Antenna/ Omniantenna
File d'attente	Queue /DropTail /PriQueue
Type d'interface	Phy/WirelessPhy
Type de canal	Channel/WirelessChannel

Tableau 4.1 Tableau des paramètres de simulation

Nous avons choisi de simuler les deux types de trafic: CBR et TCP afin de voir l'effet des modifications sur l'aspect global du nouveau protocole de routage TORA multi-interface. La génération du trafic est faite en utilisant l'utilitaire *cbrgen.tcl* fourni par NS-2. Il génère de façon aléatoire un trafic de type TCP ou CBR entre les nœuds du réseau. Pour ce faire nous lançons l'exécution de l'utilitaire *cbrgen.tcl* selon la syntaxe ci-dessous :

```
ns cbrgen.tcl [-type cbr/tcp] [-nn nodes] [-seed seed] [-mc connections][ -rate rate] > f-resultat
```

Les paramètres d'entrée de ce générateur de trafic sont définis comme suit :

- type : correspond au type de trafic soit cbr ou tcp,
- nn : représente le nombre de nœuds formant le réseau,
- nc est le nombre maximal de connexions présentes dans le réseau,

- seed est la séance génération aléatoire *random number generator*, et
- rate : représente le débit de transmission. Il est défini dans le cas trafic CBR mais pas dans le cas trafic TCP.

Pour sauvegarder le résultat d'exécution nous utilisons la directive > suivie du nom de fichier résultat. L'exécution de ce générateur donne en résultat un fichier contenant un script de trafic entre un ou plusieurs nœuds sources et un ou plusieurs nœuds destinations. Nous présentons ci-dessous un premier exemple de génération de trafic CBR suivi d'un exemple de génération de trafic TCP. L'instruction

```
ns cbrgen.tcl -type cbr -nn 10 -seed 1.0 -mc 2 -rate 2.0M > trafic-tora-udp
```

génère un fichier script pour trafic CBR entre deux connexions dans un réseau de 10 nœuds pour un débit de 2 Mbit. Le résultat sera sauvegardé dans le fichier trafic-tora-UDP. La taille des paquets transmis est de 512 octets. La deuxième instruction

```
ns cbrgen.tcl -type tcp -nn 10 -mc 2 > trafic-tora-tcp
```

génère un fichier script pour trafic TCP entre deux connexions dans un réseau de 10 nœuds. Le résultat sera sauvegardé dans le fichier trafic-tora-tcp.

Pour la mobilité des nœuds dans notre simulation, nous optons pour l'emploi de l'utilitaire setdest présent dans NS-2. Cet utilitaire génère aléatoirement le déplacement des nœuds d'une position à une autre selon une vitesse de mobilité définie comme paramètre. Aussi il est à signaler que chaque nœud peut être configuré selon un temps de pause durant lequel le nœud stoppe son mouvement, pour reprendre son déplacement à nouveau dès la fin de ce temps.

La syntaxe que nous devons écrire pour générer le fichier qui contiendra les positions initiales des nœuds du réseau et leurs différents changements de position est comme suit:

```
./setdest -v 1 -n nombrenoeds -p pausetime -M vitesseMax -t tempsSimulation -x topologieX  
-y topologieY
```

La définition des paramètres de setdest est la suivante.

- v 1 représente l'utilisation de la version 1 du générateur setdest,

- n est le nombre de nœuds mobiles qui changeront leurs positions initiales. Dans notre cas tous les nœuds du réseau sont mobiles,
- p indique le temps d'arrêt que chaque nœud doit faire avant le prochain déplacement vers sa nouvelle position. Il est mesuré en secondes,
- M est la vitesse maximale que peut atteindre un nœud durant son déplacement d'une position à une autre. La vitesse de déplacement des nœuds est variable,
- t correspond au temps total de simulation. Dans notre cas, c'est 150 secondes, et
- x, -y représentent la dimension de la topologie du réseau.

Nous citons ci-dessous un scénario de génération de mobilité utilisé dans notre simulation :

```
./setdest -v 1 -n 10 -p 0 -M 20 -t 150 -x 670 -y 670
```

Le fichier résultat contiendra les lignes des positions initiales des 10 nœuds du réseau comme par exemple

```
# nodes: 10, pause: 0,00, max speed: 20,00, max x: 670,00, max y: 670,00
```

```
$node_(0) set X_ 362,077901200904
```

```
$node_(0) set Y_ 221,761596290491
```

```
$node_(0) set Z_ 0,000000000000
```

```
$node_(1) set X_ 557,974435216441
```

```
$node_(1) set Y_ 44,564248043896
```

```
$node_(1) set Z_ 0,000000000000
```

suivies par les lignes de changement de position des 10 nœuds, comme à titre d'exemple les lignes suivantes :

```
$ns_ at 0,000000000000 "$node_(0) setdest 514,135004498549 667,061459815312  
14,309097476372"
```

```
$ns_ at 104,732799679647 "$node_(0) setdest 350,811072557397 454,332730474697  
11,361652962683"
```

Les deux lignes ci-dessus montre qu'à l'instant 0,00 seconde, le nœud 0 se déplace vers les positions x=514,13 et y=667,06 avec une vitesse de 14,30m/s. À l'instant 104,73 secondes le nœud 0 se déplace vers les positions x=350,81 et y=454,33 avec une vitesse de 11,36m/s.

À la fin du fichier nous trouvons les lignes :

```
$god_ set-dist 0 1 2
```

\$god_set-dist 0 2 1

Ces lignes servent à définir le nombre de sauts entre les nœuds 0 et 1 et les nœuds 0 et 2 qui sont de deux sauts et de un saut respectivement.

4.3.2 Métriques de performance

Pour évaluer notre algorithme de routage pour les réseaux RRCs, nous avons choisi les paramètres de performances suivants (Harmider S. et al. 2010).

Le débit moyen. C'est le nombre de paquets reçus avec succès par la destination par unité de temps. Il mesure l'efficacité d'occupation de la bande passante.

Le délai moyen de bout en bout (End to End Delay). C'est le temps moyen qu'un paquet prend pour parcourir un chemin de la source jusqu'à la destination.

Le surdébit/la charge du routage (*Routing overhead*). Il représente le nombre de paquets de contrôle transmis durant une simulation. Cette métrique évalue la consommation de la bande passante par les paquets de contrôle utilisés par le protocole de routage.

Le taux de livraison normalisé (*Packet delivery ratio*). C'est le rapport entre le nombre de paquets reçus par la destination et le nombre de paquets émis par la source au niveau agent de la couche application. Il montre l'efficacité du protocole de routage utilisé.

Le surdébit/la charge normalisé du routage (*Normalizing routing overhead*). Il représente le nombre de paquets de contrôle transmis par rapport à tous les paquets reçus.

Le nombre moyen de sauts (*average Hop count*). C'est le nombre moyen de sauts parcouru par un paquet jusqu'à sa destination.

Le taux de pertes. C'est le pourcentage des paquets envoyés et qui ne sont pas arrivés à leur destination.

CHAPITRE V

RÉSULTATS ET INTERPRÉTATION

5.1 Interprétation des résultats

Les résultats de simulation sont présentés à l'aide du logiciel Gnuplot sous forme graphique. La simulation a été faite en deux étapes. La première étape est consacrée à étudier l'apport multi-interface au protocole de routage TORA. La seconde étape consiste à développer les procédures pour adapter TORA multi-interface au protocole à radios cognitives, c'est-à-dire au concept de prioriser l'activité d'un nœud primaire par rapport aux nœuds secondaires et aussi d'interrompre les transmissions secondaires en cours lorsqu'une transmission primaire est détectée.

5.1.1 Première étape : évaluation de TORA multi-interface

Dans cette première étape nous allons analyser les résultats de simulation du protocole TORA non modifié et ceux du protocole TORA multi-interface. Nous avons fait varier le nombre d'interfaces en prenant trois interfaces ou cinq interfaces. Nous présentons dans la suite les résultats obtenus avec une analyse pour chaque résultat. Nous avons itéré les simulations une dizaine de fois et nous avons calculé la moyenne des résultats.

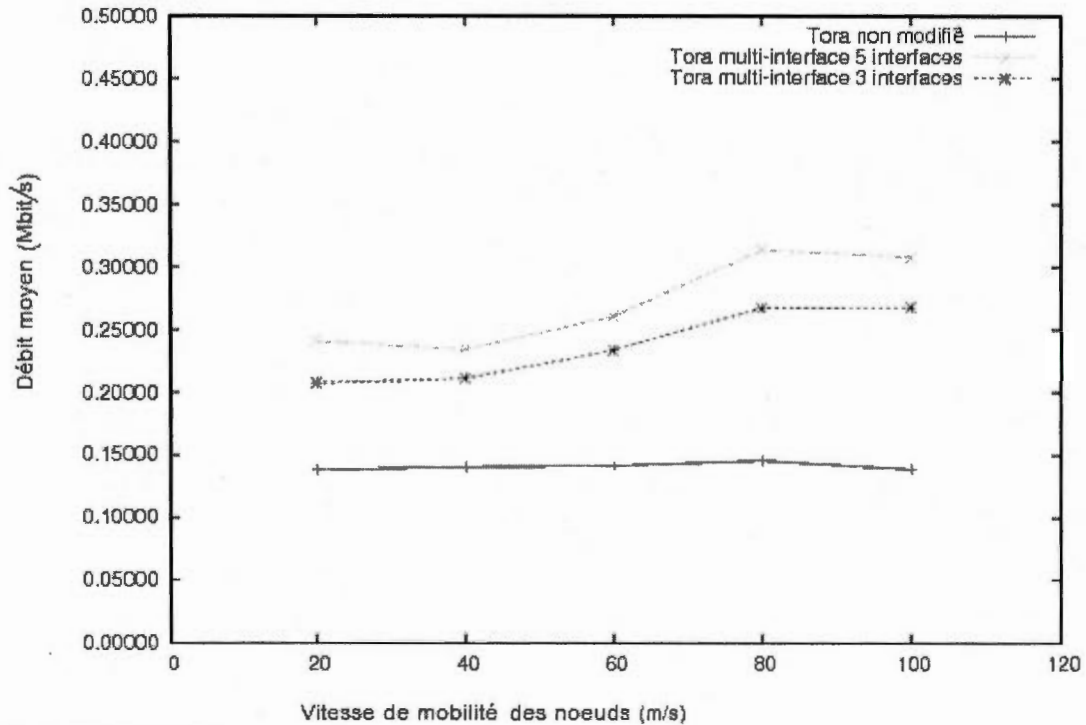


Figure 5.1 Variation du débit moyen en fonction de la vitesse de mobilité, trafic TCP

D'après la figure 5.1, nous remarquons que les valeurs de débit pour la courbe représentant une transmission utilisant trois interfaces ou cinq interfaces sont plus grandes que celles de TORA non modifié. C'est-à-dire que le débit moyen des protocoles multi-interfaces est plus élevé que le débit de TORA non modifié, ce qui répond bien à nos attentes de la modification de TORA en multi-interface, car notre but de modifier TORA est d'augmenter le débit. Ce résultat est justifié par l'augmentation de transmission de paquets durant une unité de temps. Les protocoles multi-interfaces utilisent plusieurs interfaces, chaque interface est utilisée pour envoyer des paquets. Si TORA non modifié envoie un paquet pendant une unité de temps alors TORA à trois interfaces envoie trois paquets pendant la même unité de temps et TORA à cinq interfaces envoie cinq paquets. En général, le résultat obtenu est très important mais il faut que nous observions également le débit de transmission dans le cas trafic CBR, ce que nous faisons dans la prochaine figure.

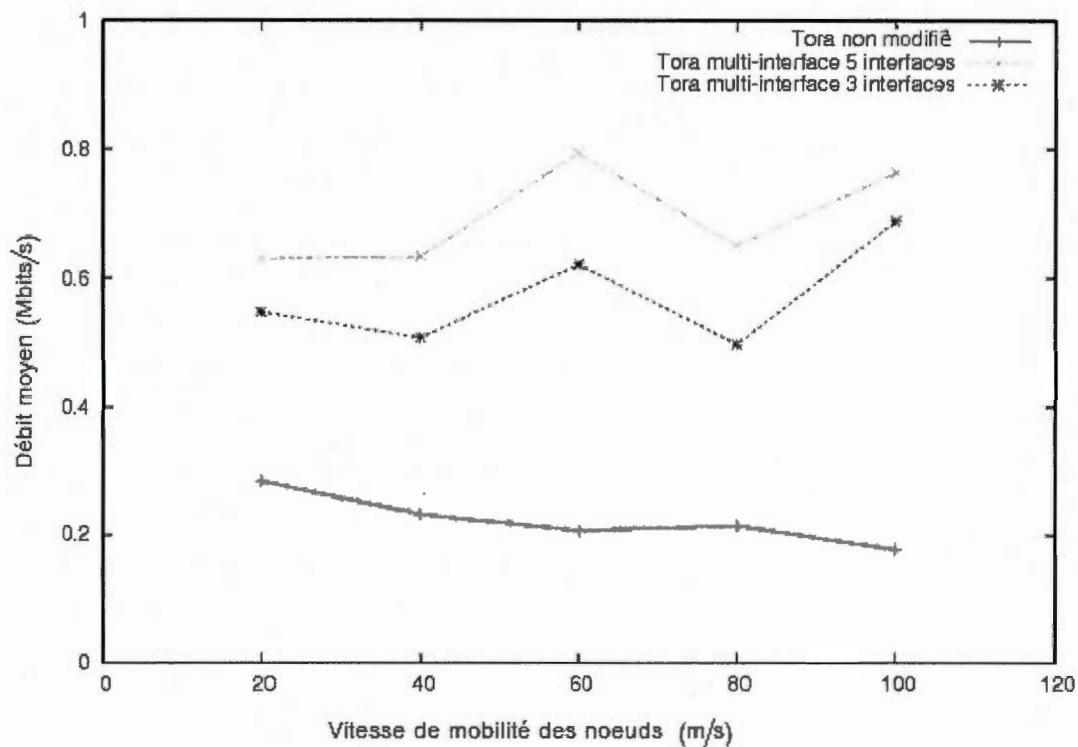


Figure 5.2 Variation du débit moyen en fonction de la vitesse de mobilité, trafic CBR

Pour le trafic CBR, nous remarquons dans la figure 5.2 que la courbe représentant le débit moyen pour une transmission utilisant trois interfaces ou cinq interfaces oscille selon la valeur de vitesse de mobilité des nœuds, ce qui n'est pas remarqué dans la figure précédente (fig. 5.1). Cette oscillation est justifiée par le fait que le trafic TCP utilise des messages d'acquittement en réponse aux paquets transmis contrairement au trafic continu CBR. Globalement le débit moyen de TORA multi-interface est plus élevé que celui du protocole TORA non modifié. Nous remarquons aussi que le débit CBR est plus élevé que le débit TCP.

Pour les deux types de trafic (TCP et CBR), il est clair que le concept multi-interface introduit une amélioration du débit moyen au protocole de routage TORA. Nous remarquons que le débit est nettement plus élevé. Nous favorisons donc l'emploi de plus d'une interface pour adapter TORA aux réseaux RRCs.

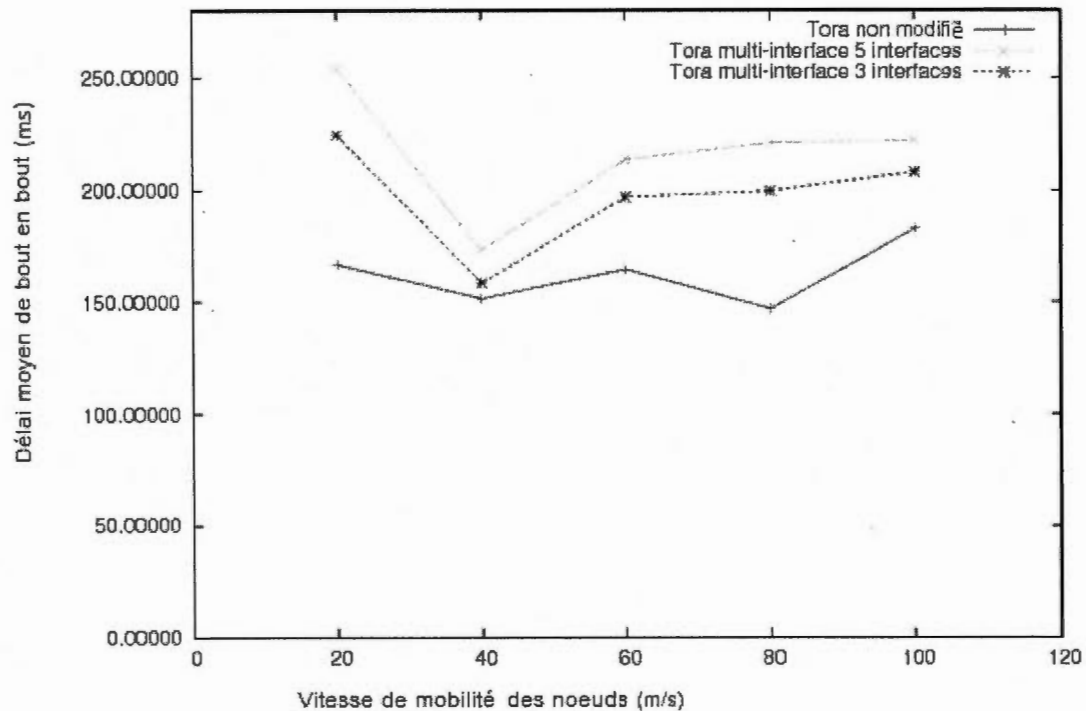


Figure 5.3 Variation du délai de bout en bout en fonction de la vitesse de mobilité, trafic TCP

Dans la figure 5.3 nous remarquons que le délai de bout en bout de TORA multi-interface est plus grand que celui du protocole TORA non modifié, pour le trafic TCP. Nous remarquons une décroissance importante du délai de bout en bout pour le protocole TORA multi-interface entre les vitesses de mobilité 20 m/s et 40 m/s, par la suite une augmentation est remarquée pour les trois protocoles. Nous déduisons que plus la vitesse de mobilité augmente plus les nœuds sont en mouvement d'où destruction fréquente de chemins qui doit être suivie par un déclenchement de processus de maintenance de route. Cela influe sur le délai d'acheminement des paquets transmis, car ils prendront plus de temps pour arriver à leur destination.

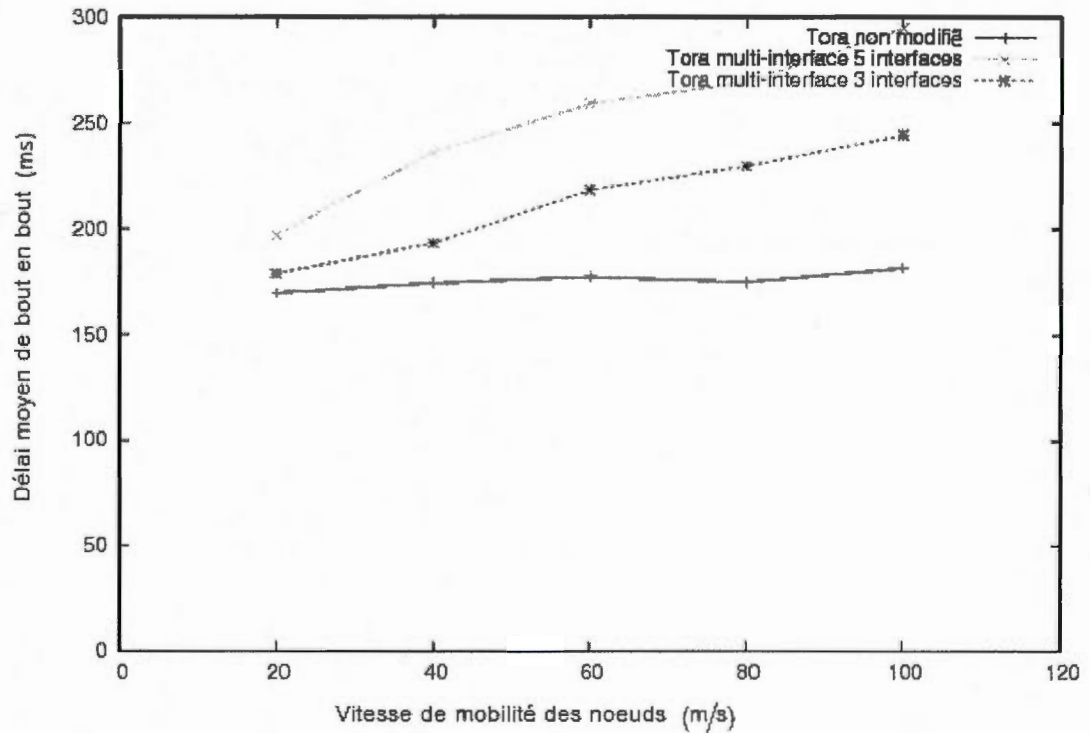


Figure 5.4 Variation du délai de bout en bout en fonction de la vitesse de mobilité, trafic CBR

Dans le cas de trafic CBR (fig. 5.4), le cas de transmission sur cinq interfaces et sur trois interfaces présente un délai élevé par rapport au protocole TORA non modifié. Ce qui revient à dire que lorsque les paquets sont transmis par TORA multi-interface, ils peuvent parcourir des chemins longs car chaque interface a un chemin, par contre TORA non modifié transmet ses paquets sur une seule interface ce qui correspond à un seul chemin qui est plus court que les chemins employés par le protocole TORA multi-interface.

Pour le délai de bout en bout qu'un paquet prend pour arriver à sa destination, nous remarquons que pour les deux types de trafic (fig. 5.3 et fig. 5.4) un paquet transmis par le protocole TORA non modifié prend moins de temps pour arriver à sa destination. Donc TORA multi-interface n'améliore pas le délai moyen de bout en bout. Cela ne pénalise pas TORA multi-interface, car le multi-interface améliore d'autres paramètres comme montré dans la suite de cette section.

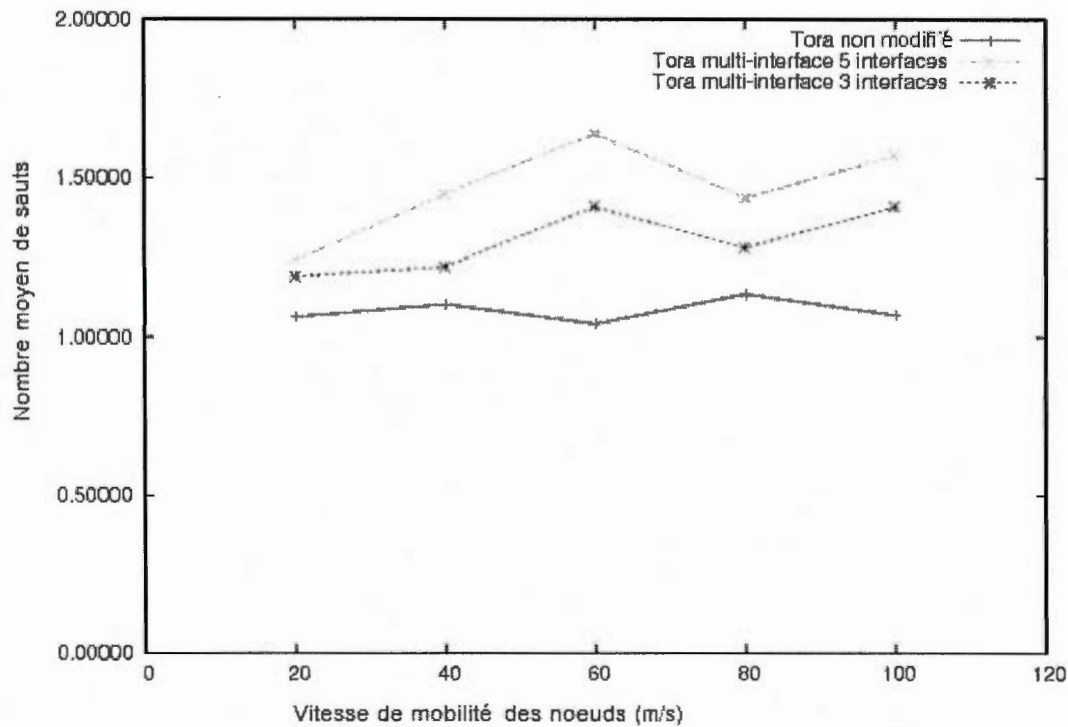


Figure 5.5 Variation du nombre moyen de sauts en fonction de la vitesse de mobilité, trafic TCP

Dans la figure 5.5 nous constatons que les protocoles multi-interfaces emploient plus de sauts que le protocole TORA non modifié. Les protocoles multi-interfaces tentent de trouver tous les chemins vers la destination, qu'ils soient court ou long, ce qui explique les valeurs élevées du nombre de sauts effectués par les paquets transmis par les protocoles TORA multi-interfaces. Un chemin court est constitué d'un seul saut car le nœud source est proche du nœud destination et aucun nœud intermédiaire n'est utilisé pour transmettre les paquets entre les deux nœuds. Par contre un chemin long est constitué de plus d'un saut, donc entre la source et la destination d'autres nœuds intermédiaires sont utilisés pour acheminer les paquets. De plus TORA non modifié exécute le processus d'optimisation dans l'établissement de chemin ce qui lui assure de court chemin. Tora multi-interface utilise tous les chemins existants pour transmettre les paquets.

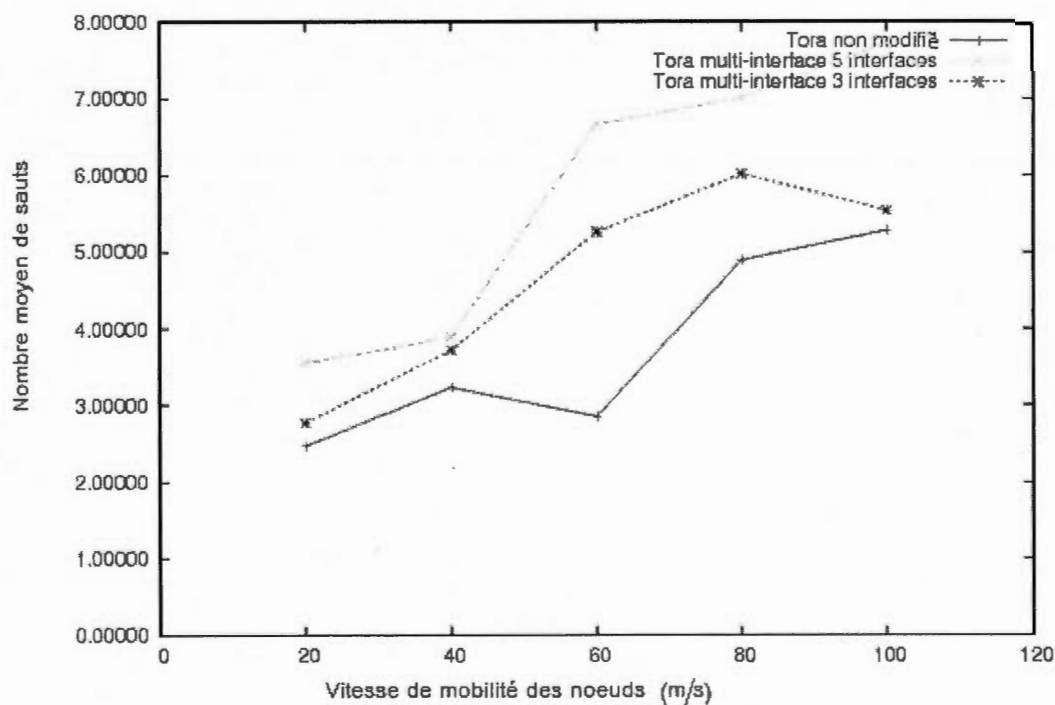


Figure 5.6 Variation du nombre moyen de sauts en fonction de la vitesse de mobilité, trafic CBR

Il est à remarquer du graphique de trafic CBR (fig. 5.6) que TORA à multi-interface emploie plus de sauts que TORA non modifié. On a le minimum à la valeur de vitesse 60 m/s qui permet de voir l'influence de la vitesse de mobilité sur la variation du nombre de sauts. Mais il est clair que les mêmes remarques précédentes (fig. 5.5) sont déduites dans ce graphique. En général, le nombre de sauts faits par les paquets envoyés par le protocole TORA à trois interfaces et à cinq interfaces est plus élevé que celui des paquets envoyés par le protocole TORA non modifié, quel que soit le type de trafic.

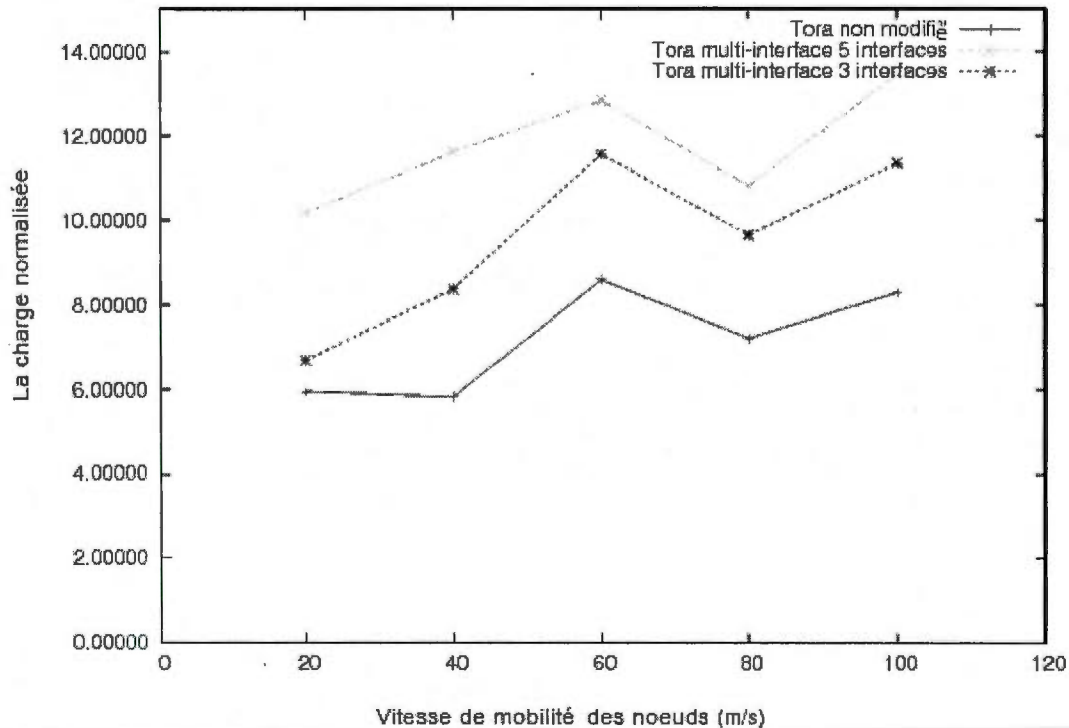


Figure 5.7 Variation de la charge normalisée en fonction de la vitesse de mobilité, trafic TCP

Le nombre de paquets envoyés durant la simulation (fig. 5.7) est plus élevé lorsque le protocole TORA emploie cinq interfaces ou trois interfaces contrairement au cas du protocole TORA non modifié. Durant la transmission, plus la vitesse de mobilité des nœuds augmente, plus la charge normalisée du réseau augmente selon l'aspect général des graphiques, malgré les légères exceptions à certains points. La charge normalisée du réseau varie car le nombre de paquets de contrôle change selon le nombre de sauts entre la paire source-destination. Si le nœud destination se trouve proche du nœud source, il y aura moins de sauts. Les paquets d'établissement de chemin seront moins nombreux et par conséquent la charge du réseau diminue aussi.

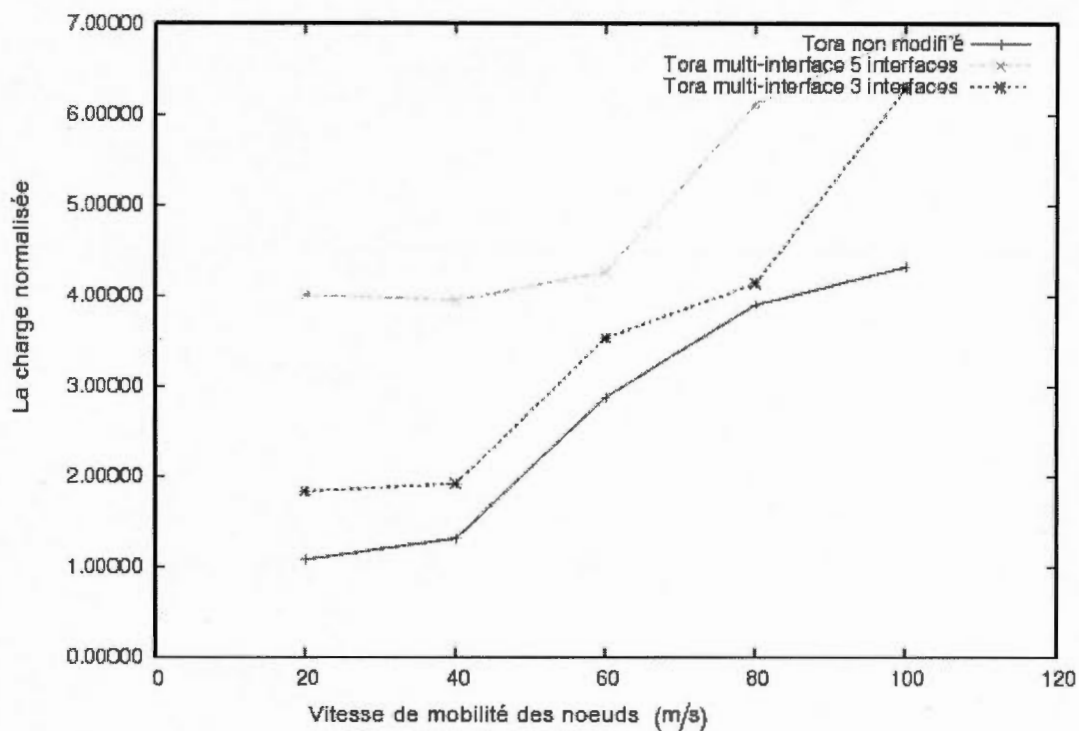


Figure 5.8 Variation de la charge normalisée en fonction de la vitesse de mobilité, trafic CBR

Quand le trafic CBR est considéré (fig. 5.8), nous retrouvons les mêmes observations que pour la figure 5.7. Une croissance de la charge normalisée du réseau est constatée pour chaque protocole. La justification est que plus la vitesse de mobilité augmente plus les nœuds se déplacent, provoquant ainsi la rupture de chemins. Des paquets de contrôles sont émis par le protocole durant le processus de maintenance de route ce qui rajoute des paquets au flux du réseau. Dans l'ensemble des cas, le protocole TORA non modifié a une charge plus petite que les deux autres. Nous pouvons expliquer cela du fait que nous envoyons plus de paquets de contrôle sur plus d'une interface, nous multiplions le nombre de paquets de routage par le nombre d'interfaces présentes sur les nœuds.

Dans les deux types de trafic, nous remarquons que la charge normalisée du réseau est variable. Selon sa définition, elle dépend du nombre de paquets de routage par rapport aux paquets reçus. Pour mieux comprendre cette variation, nous allons analyser la charge du réseau dans les figures qui suivent, sans tenir compte des paquets reçus.

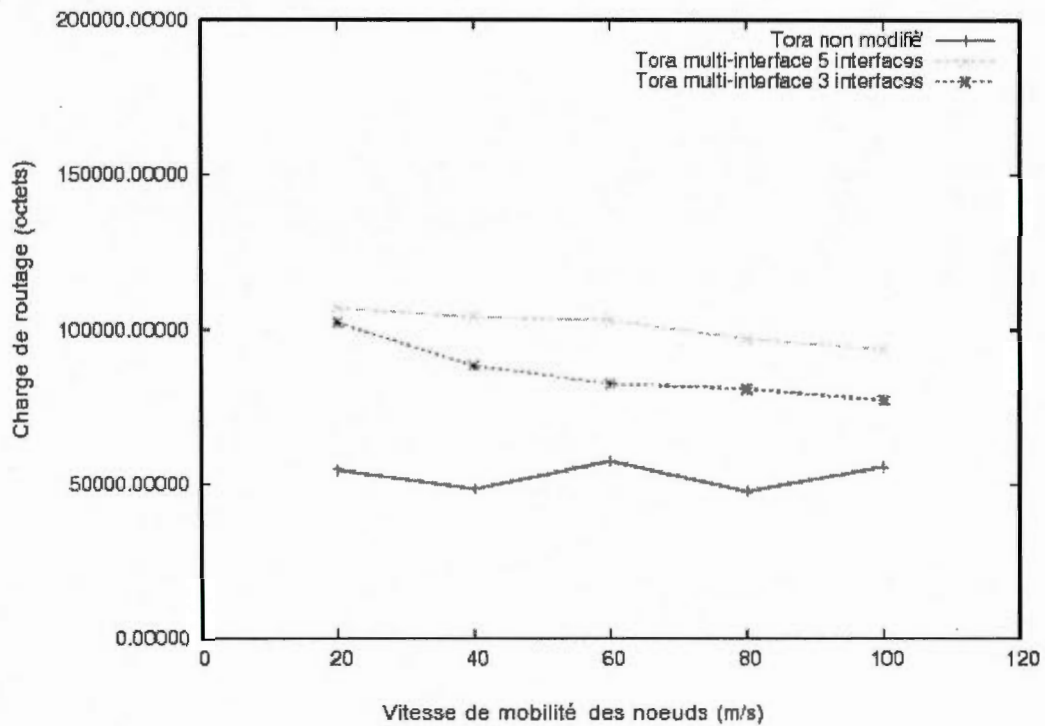


Figure 5.9 Variation de la charge de routage en fonction de la vitesse de mobilité, trafic TCP

Selon la figure 5.9, il est clair que le nombre de paquets de routage transmis par le protocole de routage employant trois interfaces et celui à cinq interfaces est nettement plus élevé que celui transmis par le protocole TORA non modifié. L'analyse de ce graphique confirme ce que nous avons déduit concernant la charge normalisée de routage à la figure 5.8.

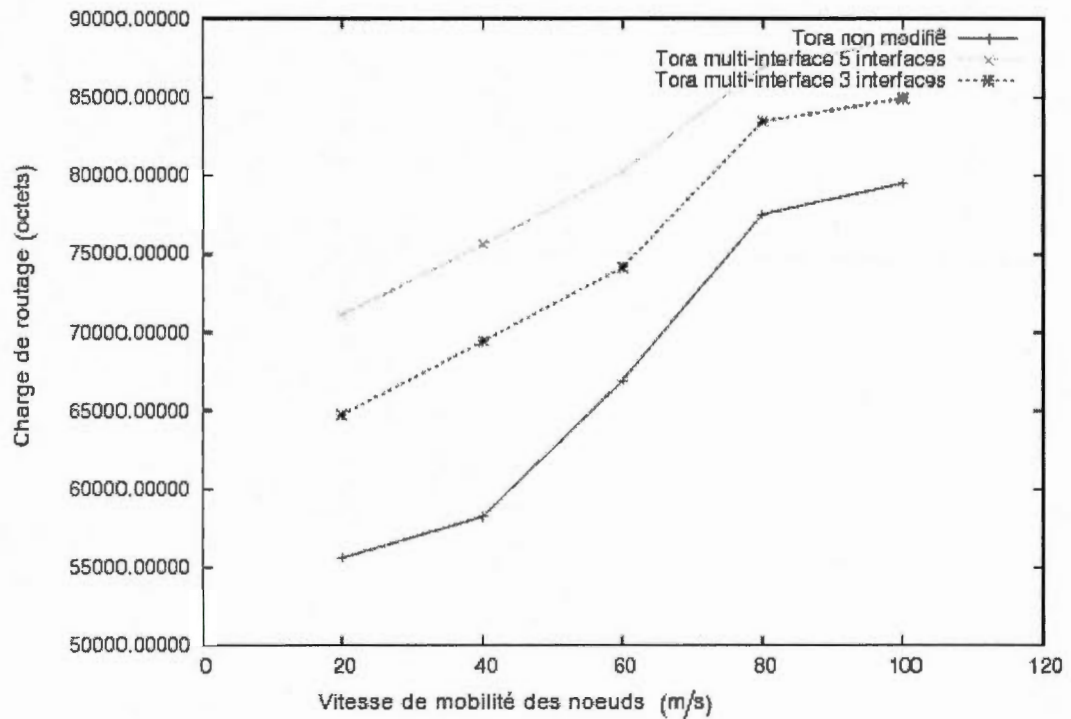


Figure 5.10 Variation de la charge en fonction de la vitesse de mobilité, trafic CBR

Pour le trafic CBR (fig. 5.10), nous remarquons presque la même allure de courbe que précédemment. La charge du réseau pour TORA non modifié a moins de paquets de routage à transmettre contrairement à TORA multi-interface. Ce résultat démontre que plus nous utilisons de points de communication pour un nœud, plus nous envoyons des paquets de contrôle.

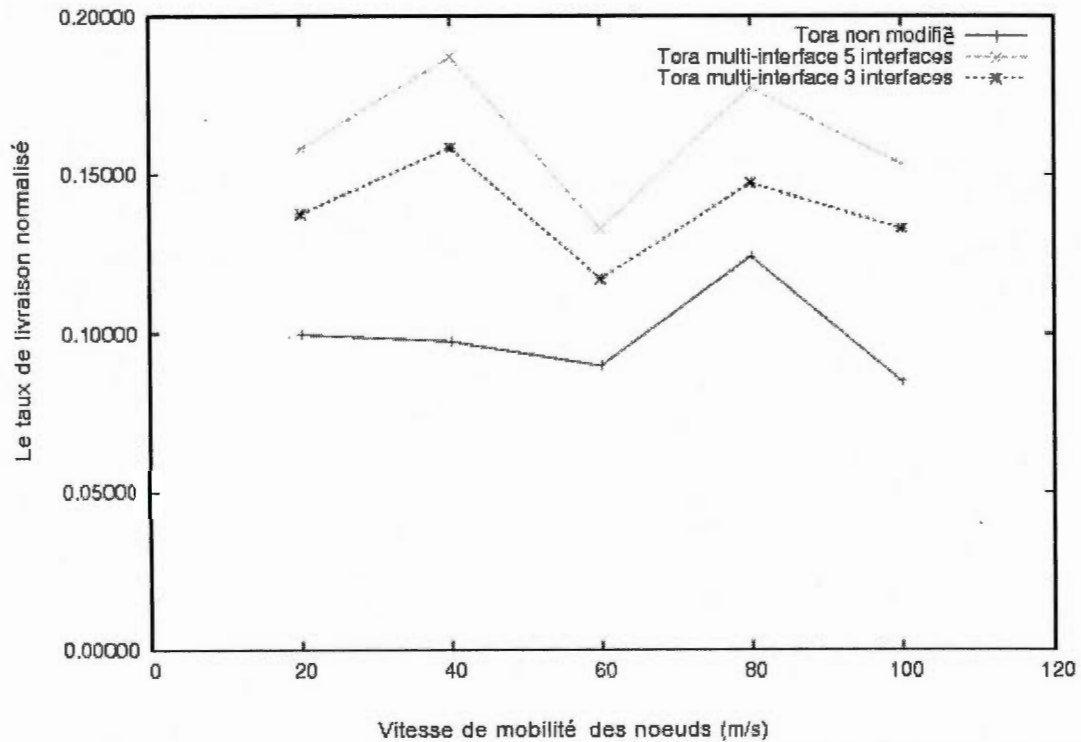


Figure 5.11 Variation du taux de livraison normalisé en fonction de la vitesse de mobilité, trafic TCP

Nous remarquons que les courbes qui représentent les paquets transmis et arrivés à leur destination par les protocoles de routage étudiés, présentent une diminution du taux de livraison pour la valeur de vitesse 100m/s. Pour les protocoles TORA à trois-interfaces et TORA à cinq interfaces nous remarquons que les courbes de taux de livraison normalisé sont plus élevées que celle de TORA non modifié.

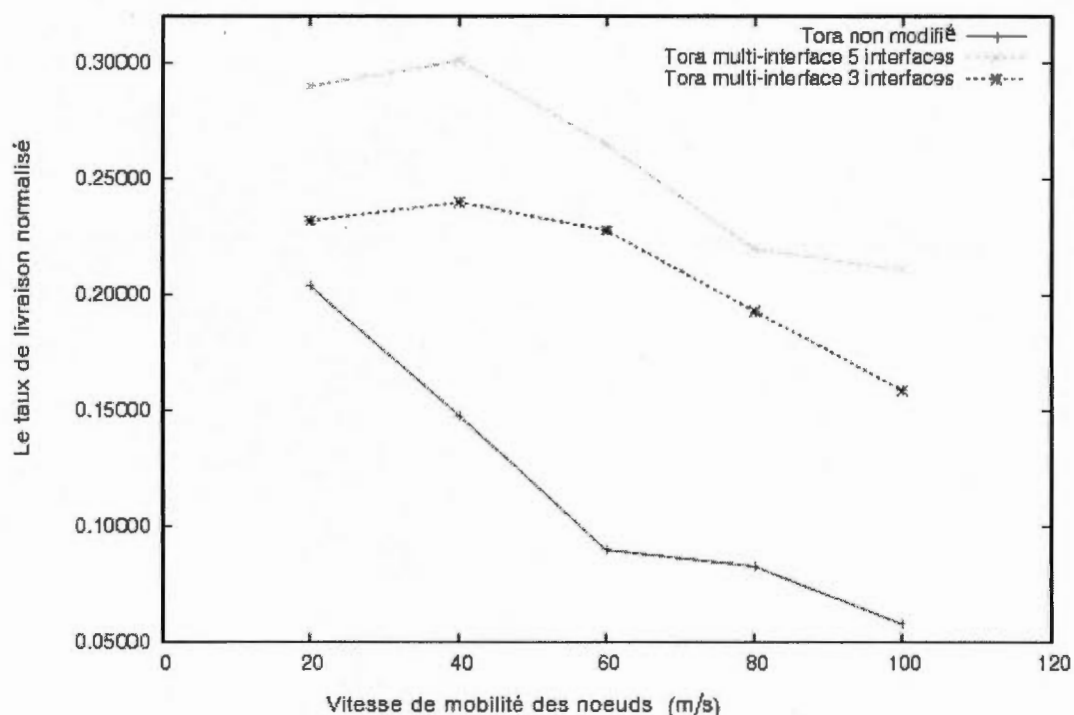


Figure 5.12 Variation du taux de livraison normalisé en fonction de la vitesse de mobilité, trafic CBR

Pour le trafic CBR comme montré dans la figure 5.12, nous voyons que la variation du taux de livraison de paquet (PDR) est très variable. Pour toutes les autres valeurs de vitesse de mobilité, TORA à cinq interfaces présente un meilleur taux par rapport aux deux autres protocoles. Pour le cas de trafic CBR, l'explication de la diminution du taux de livraison est dûe à la perte de paquets transmis durant la mobilité des nœuds, vu l'augmentation de la vitesse de mobilité. Alors nous concluons que TORA non modifié est moins performant que les protocoles à trois interfaces et à cinq interfaces pour le taux de livraison normalisé.

En conséquence et d'après la définition du taux de livraison de paquets qui est un moyen de mesurer l'efficacité d'un protocole de routage, nous pouvons dire que le protocole TORA multi-interface est plus efficace que le protocole TORA non modifié que ce soit pour le trafic TCP ou le trafic CBR.

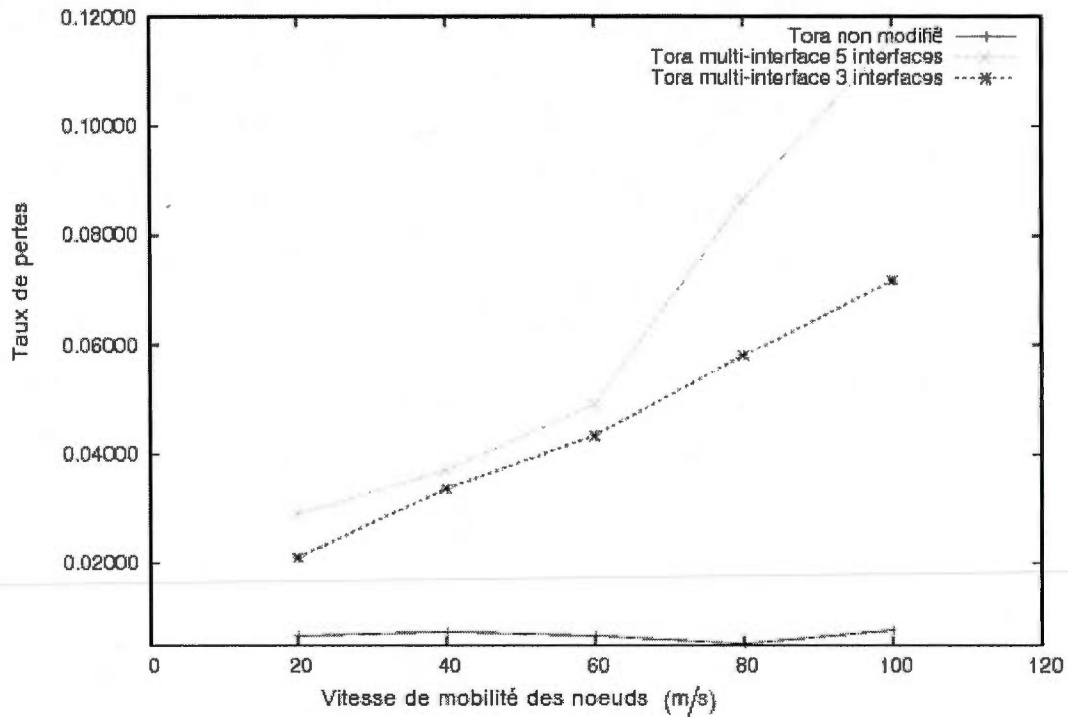


Figure 5.13 Variation du taux de pertes en fonction de la vitesse de mobilité, trafic CBR

Le taux de pertes calculé pour TORA non modifié est moins élevé que celui de TORA multi interface. La justification est que TORA non modifié envoie moins de paquets sur une seule interface entre les nœuds du réseau comparativement à l'envoi de paquets sur plusieurs interfaces par chaque nœud qui engendre plus de possibilité de collisions et donc de pertes.

En conclusion, malgré la variation de la vitesse de mobilité des nœuds, il est clair que le multi interface apporte beaucoup d'amélioration au protocole TORA. Aussi nous signalons que la variation de vitesse de mobilité a été introduite dans notre simulation afin de nous

aider à s'approcher d'un environnement opportuniste où le changement des activités des nœuds primaires et des nœuds secondaires est très fréquent.

Pour avoir une vue plus détaillée de notre modification du protocole de routage TORA, nous avons fait d'autres simulations mais cette fois ci, nous avons varié le temps de pause (*pause time*). Les mêmes paramètres sont utilisés pour les deux types de trafic TCP et CBR. Dans ce cas la vitesse de mobilité est fixée à 20 m/s. Les résultats obtenus sont représentés dans les figures suivantes.

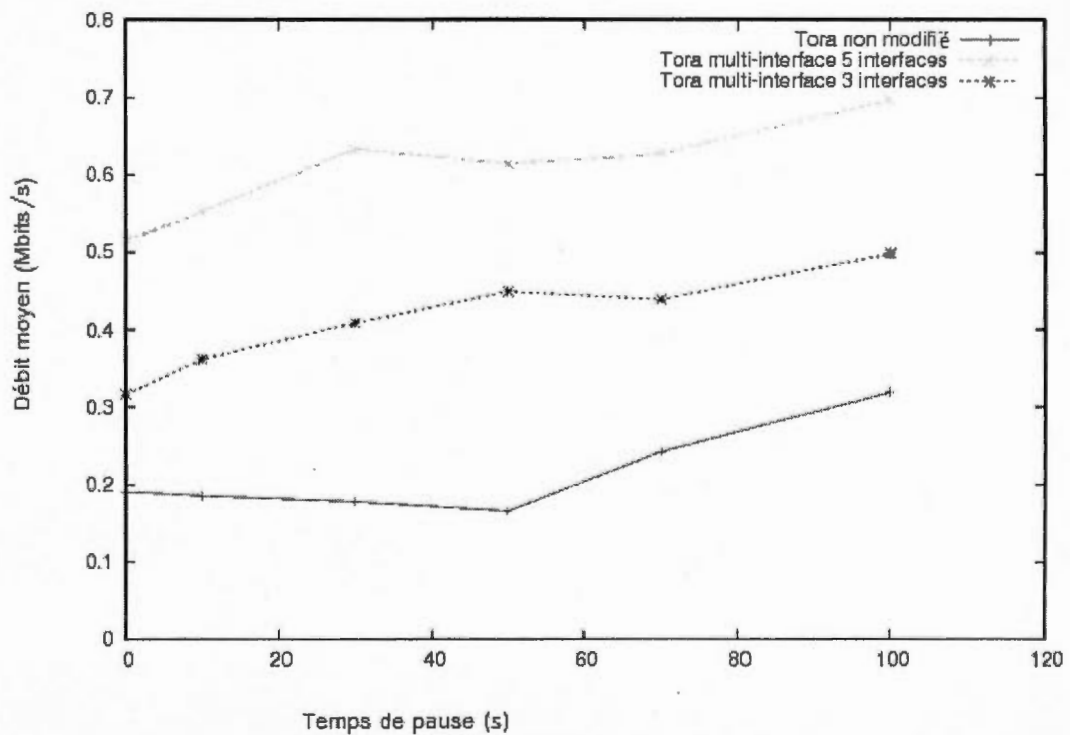


Figure 5.14 Variation du débit moyen en fonction de temps de pause des nœuds, trafic TCP

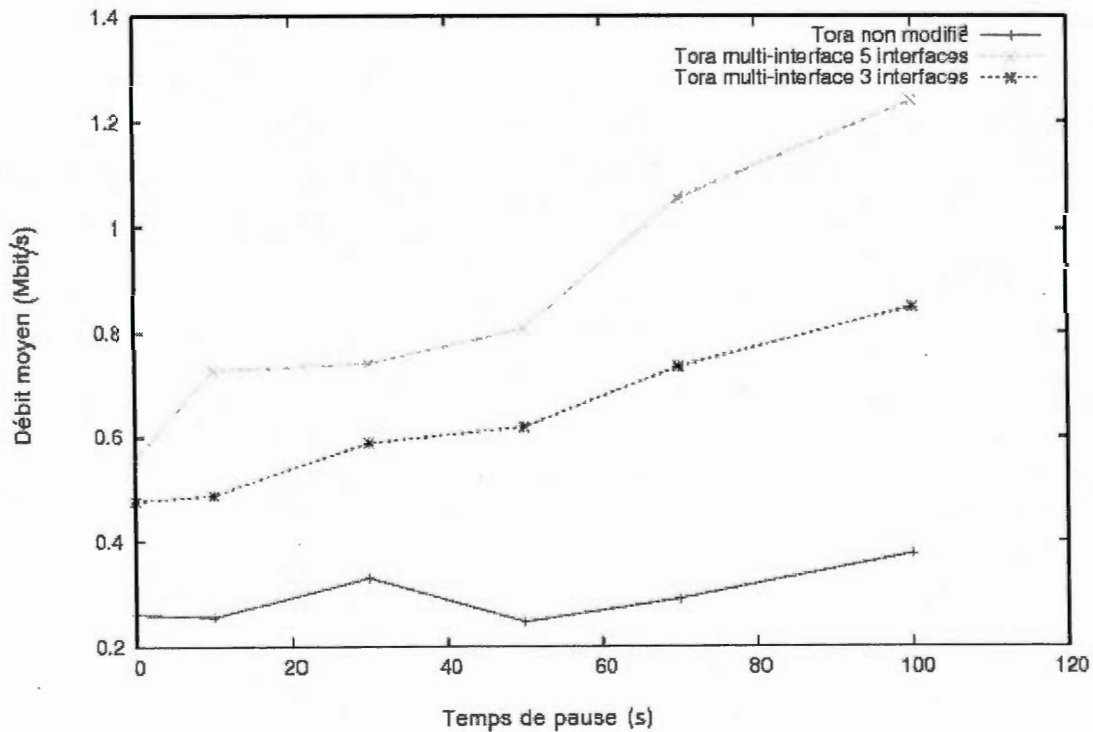


Figure 5.15 Variation du débit moyen en fonction de temps de pause des nœuds, trafic CBR

Le débit moyen du protocole multi-interface est plus élevé que TORA non modifié dans le cas de trafic TCP et le cas de trafic CBR (fig. 5.14 et fig. 5.15). Cela confirme ce que nous avons constaté dans la simulation avec variation de la vitesse de mobilité.

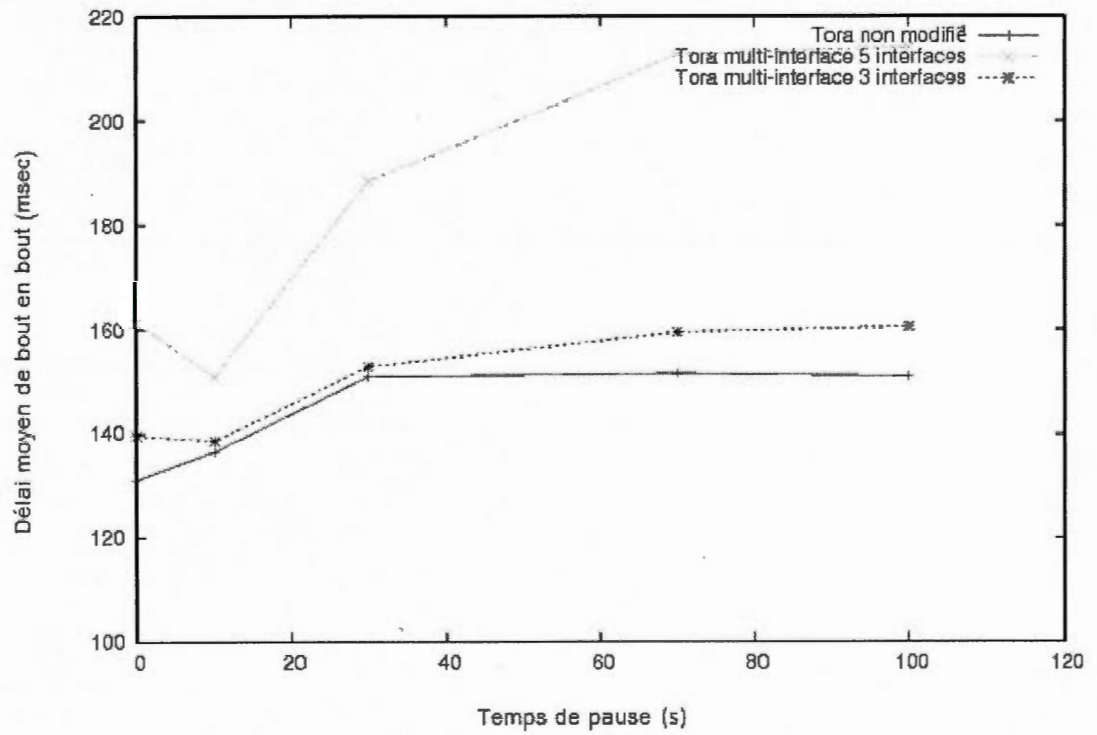


Figure 5.16 Variation du délai de bout en bout en fonction de temps de pause des nœuds, trafic TCP

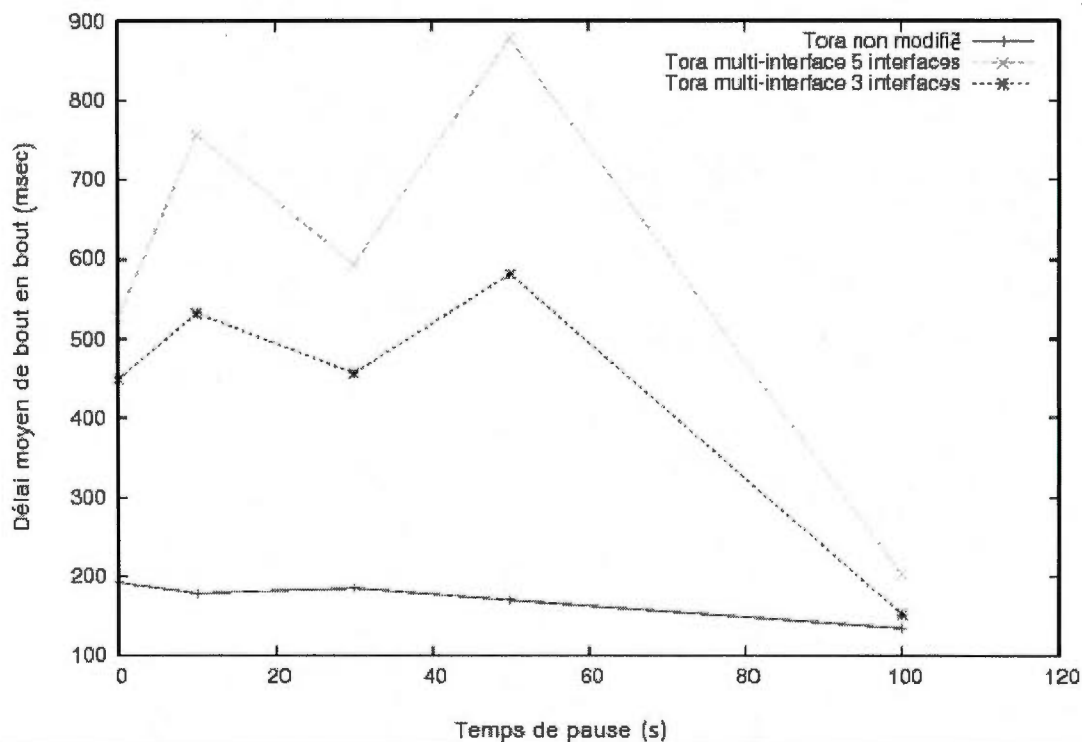


Figure 5.17 Variation du délai de bout en bout en fonction de temps de pause des nœuds, trafic CBR

Le délai de bout en bout de TORA non modifié est plus petit que celui de TORA multi- interfaces. Cela ne pénalise en aucun cas le multi-interface car l'utilisation de plusieurs interfaces peut causer une augmentation de temps de propagation des paquets sur un chemin long si le processus d'optimisation de route ne trouve pas un chemin plus court (fig. 5.16 et fig. 5.17).

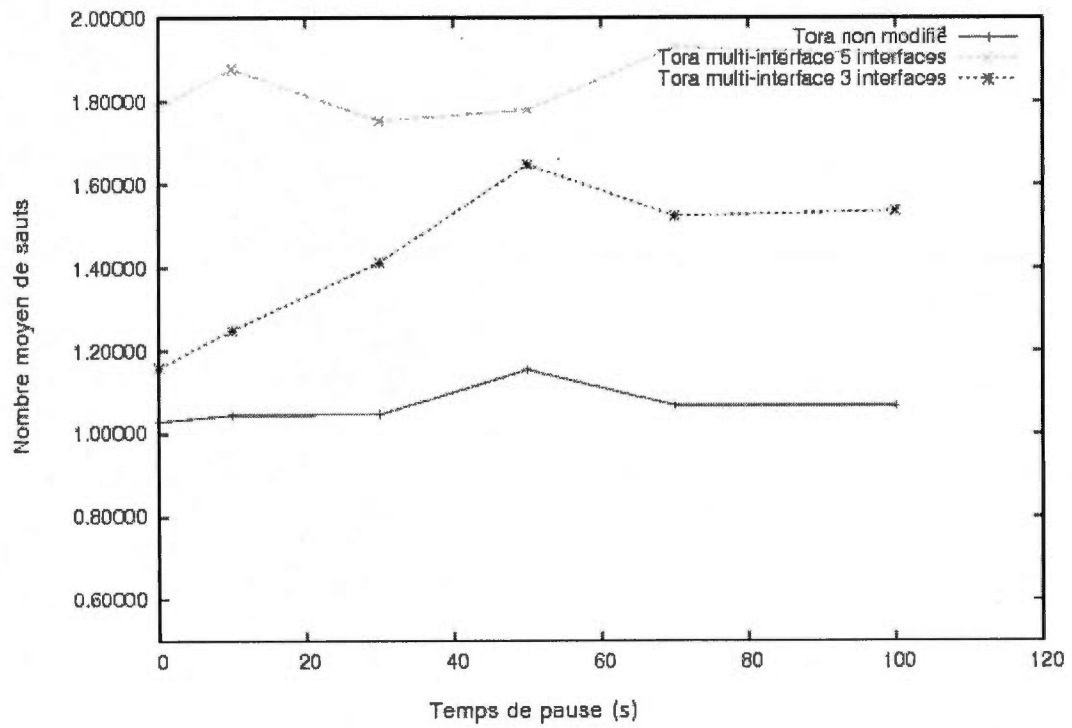


Figure 5.18 Variation du nombre moyen de sauts en fonction de temps de pause des nœuds, trafic TCP

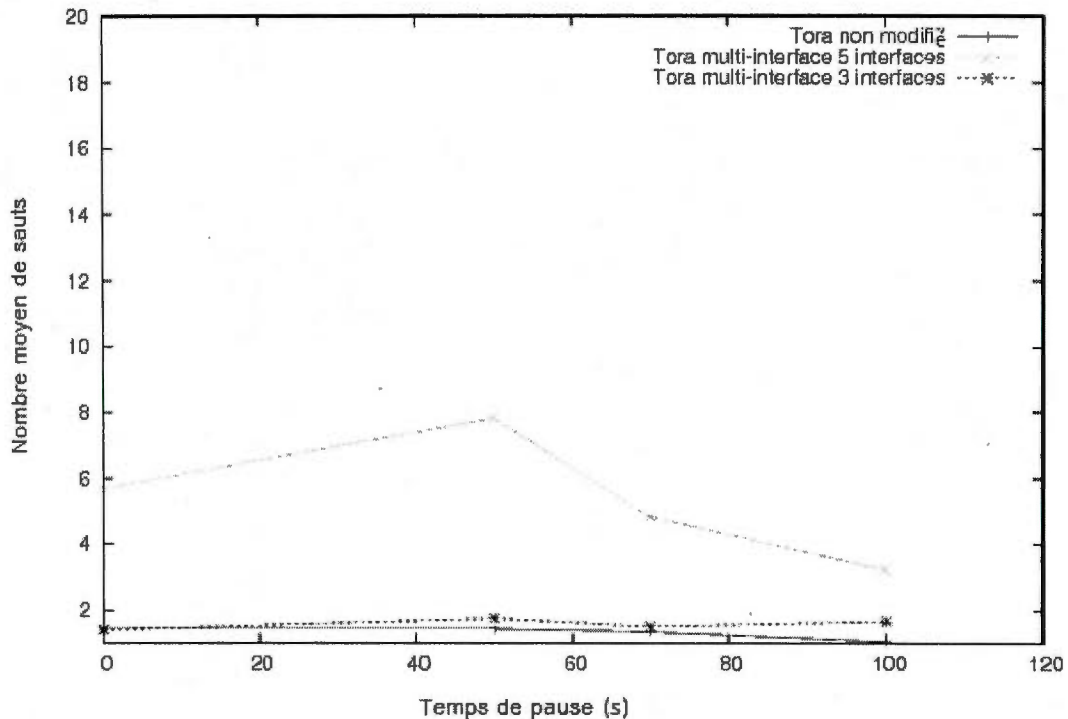


Figure 5.19 Variation du nombre moyen de sauts en fonction de temps de pause des nœuds, trafic CBR

Pour le nombre moyen de sauts qu'un paquet parcourt jusqu'à sa destination, nous remarquons que les courbes des trois protocoles se rapprochent, ce qui signifie que c'est presque le même nombre de sauts que fait chacun d'eux, sauf pour la valeur 50 s où un pic (fig. 5.19) d'une valeur supérieure à 6 sauts est fait par le protocole à cinq interfaces. Les deux graphiques (fig. 5.18 et fig. 5.19) représentant le trafic TCP et le trafic CBR nous donnent la même conclusion que l'analyse faite dans le cas de la simulation avec variation de vitesse de mobilité. Les paquets des protocoles multi-interfaces ont plus de sauts à faire que les paquets du protocole TORA non modifié.

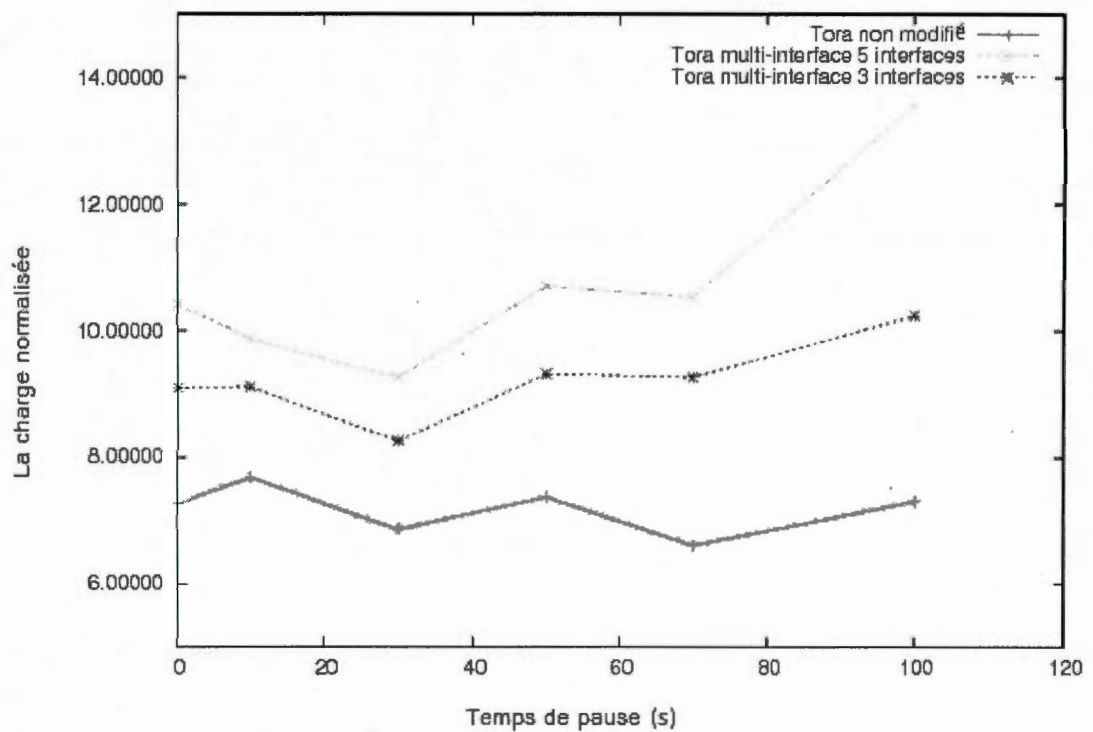


Figure 5.20 Variation de la charge normalisée en fonction de temps de pause des nœuds, trafic TCP

Dans la figure 5.20 TORA non modifié a nettement moins de charge normalisée que TORA multi-interface pour un trafic TCP. Pour un trafic CBR (fig. 5.21) la charge normalisée est très variable pour les trois protocoles. Dans le cas général, il est clair que les protocoles multi-interfaces ont plus de charge car ils transmettent les données sur plusieurs interfaces. En regardant les figures dans la suite de cette section, nous pouvons conclure que la charge du réseau augmente si le protocole utilise plus d'une interface pour acheminer les paquets. Pour cela, nous avons revu la charge du réseau en prenant en considération que les paquets de routage comme dans le cas de variation de vitesse de mobilité.

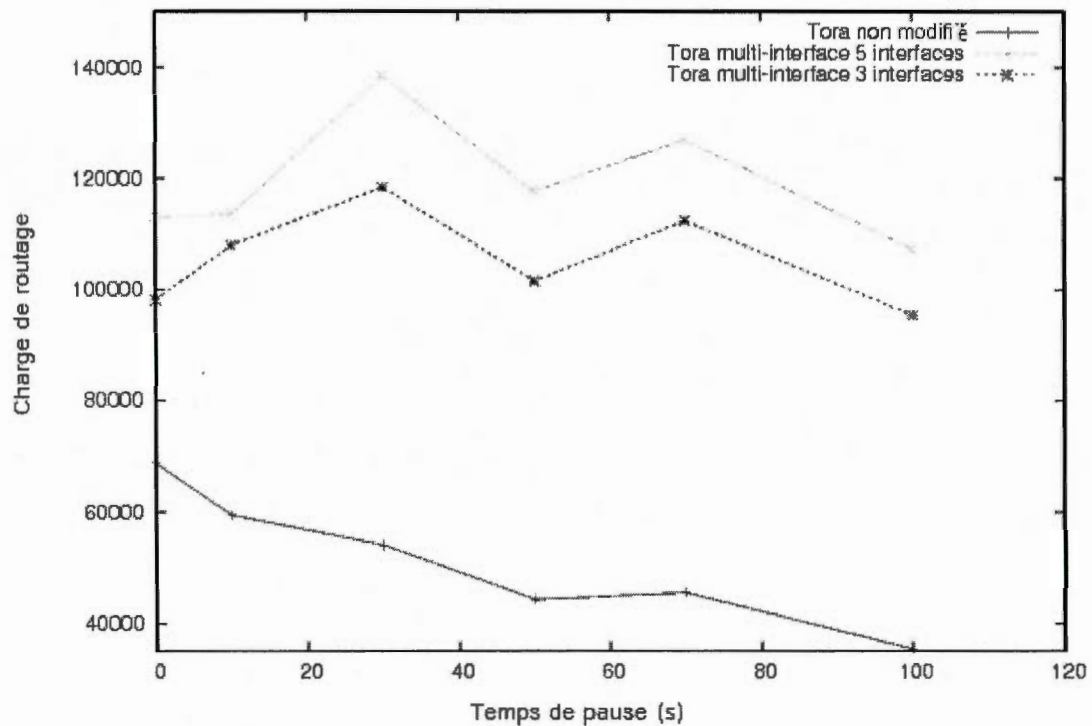


Figure 5.21 Variation de la charge du réseau en fonction de temps de pause des nœuds, trafic CBR

Dans la figure 5.21, la charge du réseau est élevée dans le cas de TORA multi-interface car chaque interface transmet des paquets de contrôle pour établir les chemins entre les nœuds du réseau. Par contre TORA non modifié transmet moins de paquets de contrôle dans l'établissement de routes vu qu'il emploie une seule interface. Ce résultat nous permet de conclure que l'emploi de plus d'une interface nous met devant un problème de la charge du réseau que nous proposons comme perspective afin d'améliorer le protocole TORA multi-interface dans le futur.

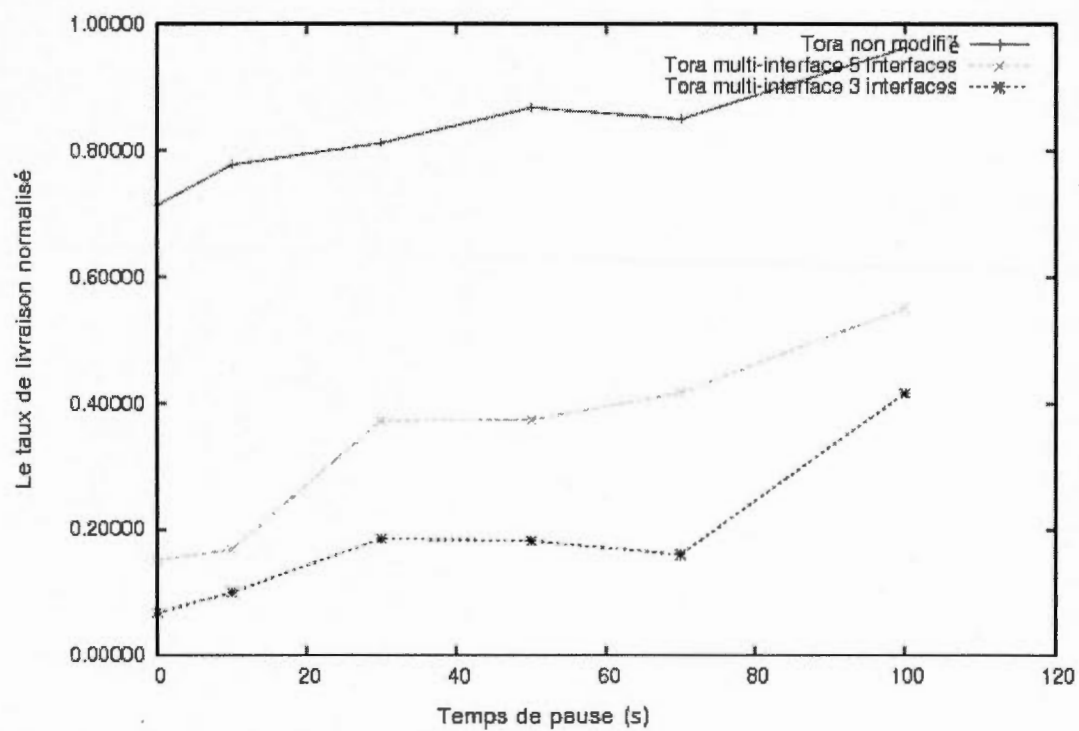


Figure 5.22 Variation du PDR en fonction de temps de pause des nœuds, trafic TCP

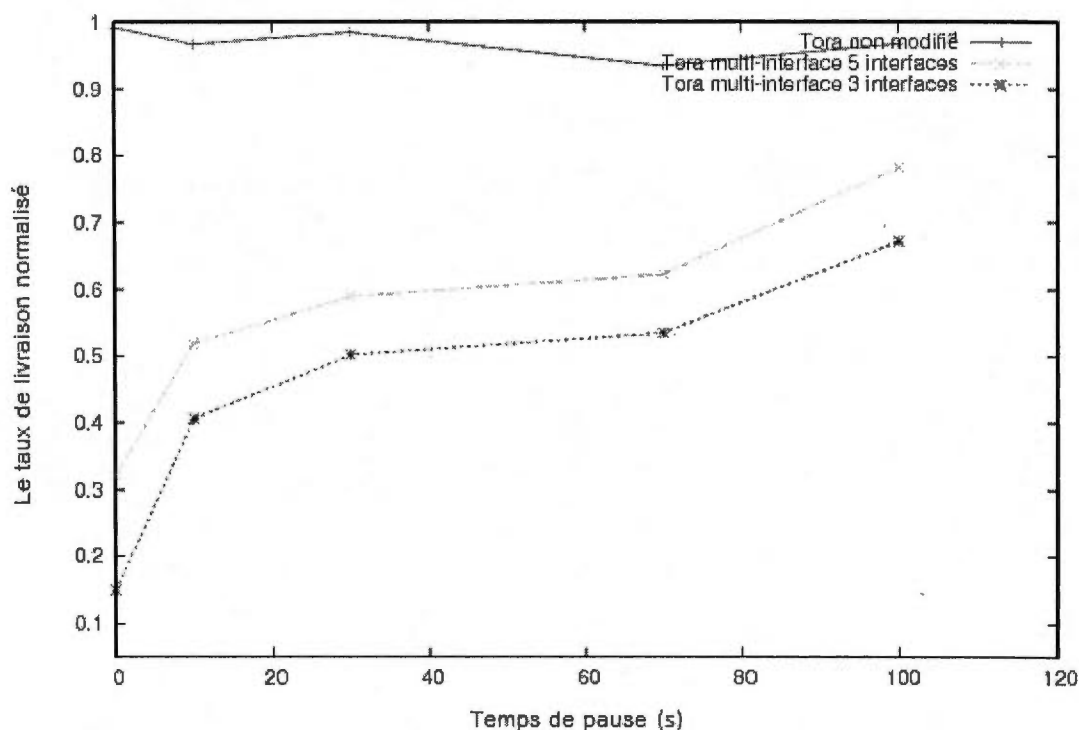


Figure 5.23 Variation du taux de livraison normalisé (PDR) en fonction de temps de pause des nœuds, trafic CBR

Le taux de livraison de TORA non modifié est plus grand que celui de TORA multi-interface, ce qui favorise l'emploi de plus d'une interface afin d'assurer une transmission de paquets sans avoir beaucoup de pertes. Presque tous les paquets transmis seront reçus selon la définition de taux de livraison de paquets (fig. 5.22 et fig. 5.23). Contrairement au trafic TCP, le trafic CBR est dépourvu d'un mécanisme d'acquittement de message après réception de paquets, par conséquent dans le cas de perte de paquets il n'y a pas de moyen à les retransmettre. Cette perte non corrigée explique la diminution du taux de livraison normalisé. Aussi nous remarquons que lorsque le temps de pause augmente c'est-à-dire que les nœuds ne changent pas leurs positions actuelles pendant un intervalle de temps assez grand, le taux de livraison augmente aussi. Lorsque les nœuds du réseau ne sont pas en mouvement alors les routes établies ne sont pas rompues et les transmissions s'effectuent sans coupure.

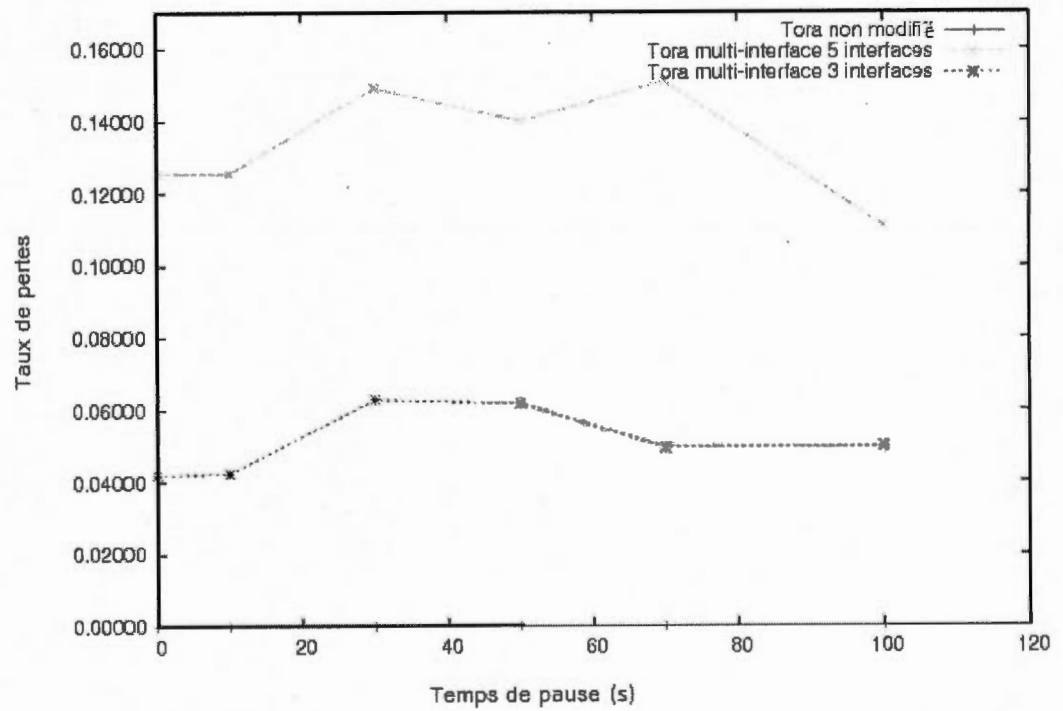


Figure 5.24 Variation du taux de pertes en fonction de temps de pause des nœuds, trafic TCP

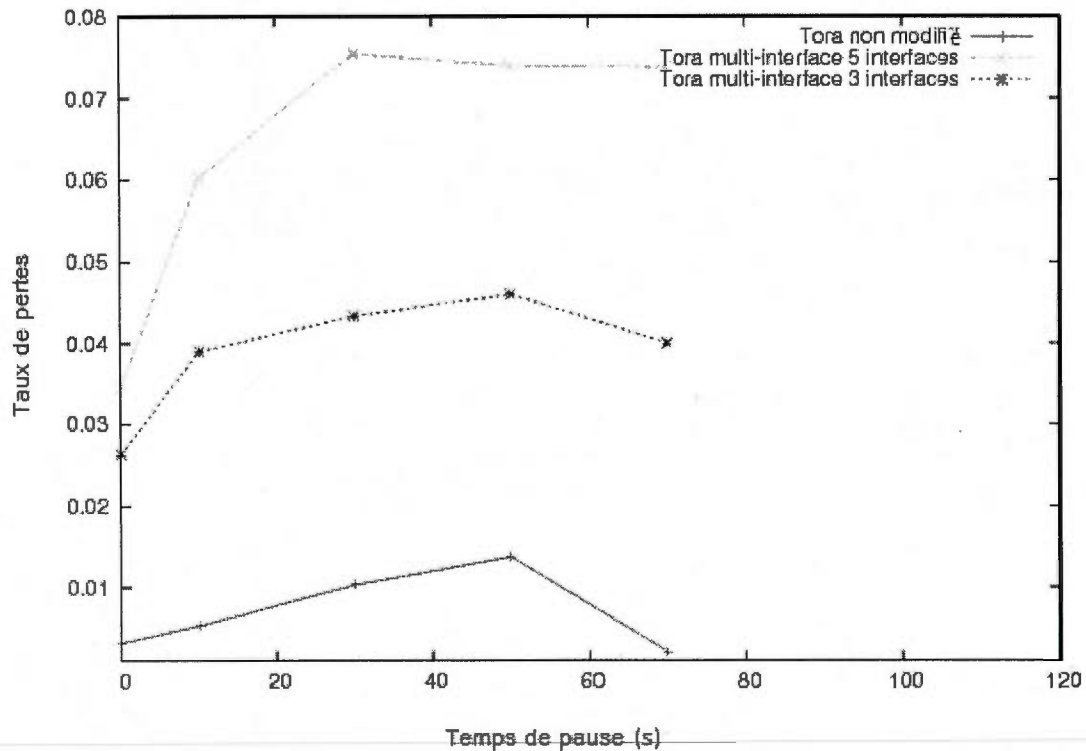


Figure 5.25 Variation du taux de pertes en fonction de temps de pause des nœuds, trafic CBR

Le taux de pertes signalé pour TORA multi-interface n'est pas grande. Elle est aux alentours de 10^{-2} . Même pour TORA non modifié elle est très réduite et il n'y a pas beaucoup de différence entre les deux protocoles vis-à-vis le taux de pertes.

À la fin de cette première étape de simulation, nous pouvons faire les constatations suivantes. De façon similaire à ce qui a été fait et justifié au début de cette simulation, nous pouvons dire que la variation de la vitesse de mobilité ou la variation de temps de pause des nœuds nous a donné presque les mêmes résultats. Ces résultats démontrent que la modification de TORA pour supporter le multi-interface est très avantageuse. Elle nous permet de proposer une première solution pour adapter TORA aux réseaux RRCs. Pour avoir

des courbes assez lisses que celles présentent dans cette section, nous proposons d'augmenter le nombre d'itérations des simulations aux alentours de 100 itérations.

5.1.2 Deuxième étape : adaptation au réseau à radios cognitives

Dans cette partie nous appliquons une modification sur les paquets utilisés par l'algorithme TORA pour l'établissement ou la suppression de chemin. Nous avons mis une valeur égale à 1 dans le champ "Réservé" des paquets TORA pour identifier le nœud secondaire. Pour le nœud primaire, nous avons laissé le champ inchangé. Les nœuds primaires sont présentés en bleu dans les figures générées par l'outil de visualisation dit "Nam" de NS-2. Au début de la simulation, les nœuds primaires n_{10} et n_{11} sont loin de la zone d'activité des nœuds secondaires n_4 et n_7 .

Le processus de découverte de chemin est lancé en premier. Dans la figure 5.29, ce processus est visible par les cercles autour de chaque nœud du réseau. Ces cercles sont des paquets de requête d'établissement de chemin QRY. Comme il apparaît sur la figure 5.29 des cercles sortant de n_4 touchent les nœuds qui lui sont proches. Ce sont ces voisins directs. Ces nœuds émettent à leur tour des paquets QRY (cercle au niveau du nœud n_1) à leurs voisins directs jusqu'à l'arrivée de ce paquet à destination n_7 . Le processus est détaillé dans la section création de route du protocole TORA au chapitre III.

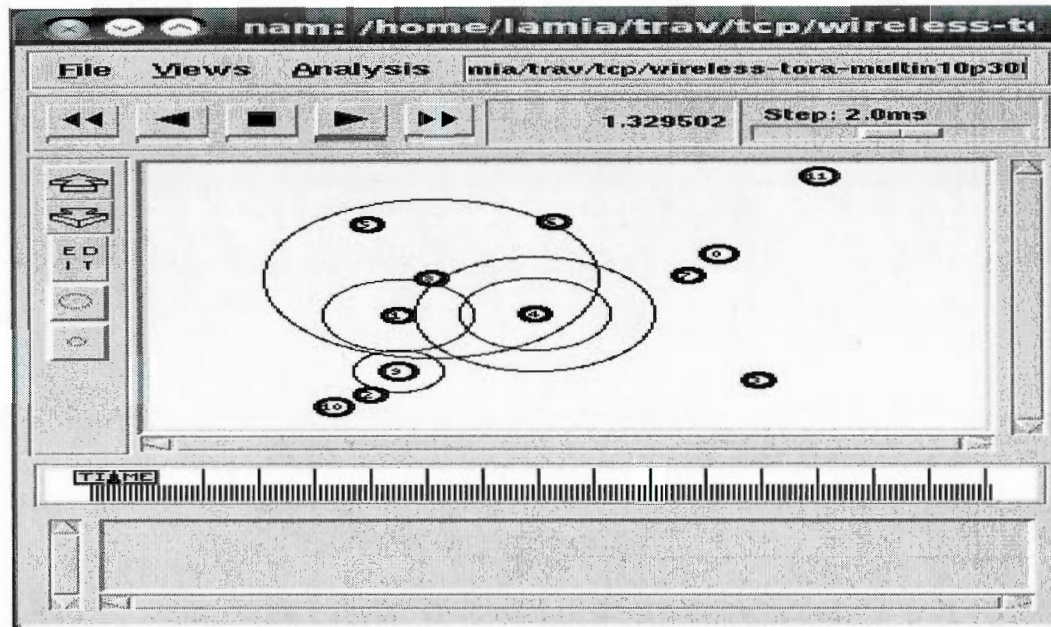


Figure 5.26 Processus de découverte et d'établissement de chemin entre nœuds secondaires

Une fois que le trafic a commencé par les deux nœuds secondaires n_4 et n_7 , les nœuds primaires n_{10} et n_{11} en bleu s'approchent de la zone d'activité de ces deux nœuds secondaires. Dès que le temps est supérieur ou égale à 15 s comme indiqué sur la partie supérieure de l'outil de visualisation de NS-2 "NAM" (encerclé en rouge dans la figure 5.30), les deux nœuds primaires sont assez proches de la zone d'activité des nœuds secondaires. Des messages de contrôles sont échangés entre les nœuds primaires n_{10} et n_{11} . Cet échange influe sur la communication entre les nœuds secondaires. Les cercles sortant de n_{10} (représentent l'envoi de paquets de contrôle QRY) touchent directement n_4 , d'où des paquets d'un nœud primaire n_{10} arrivent à un nœud secondaire n_4 . Le nœud n_4 doit réagir et interrompre sa transmission.

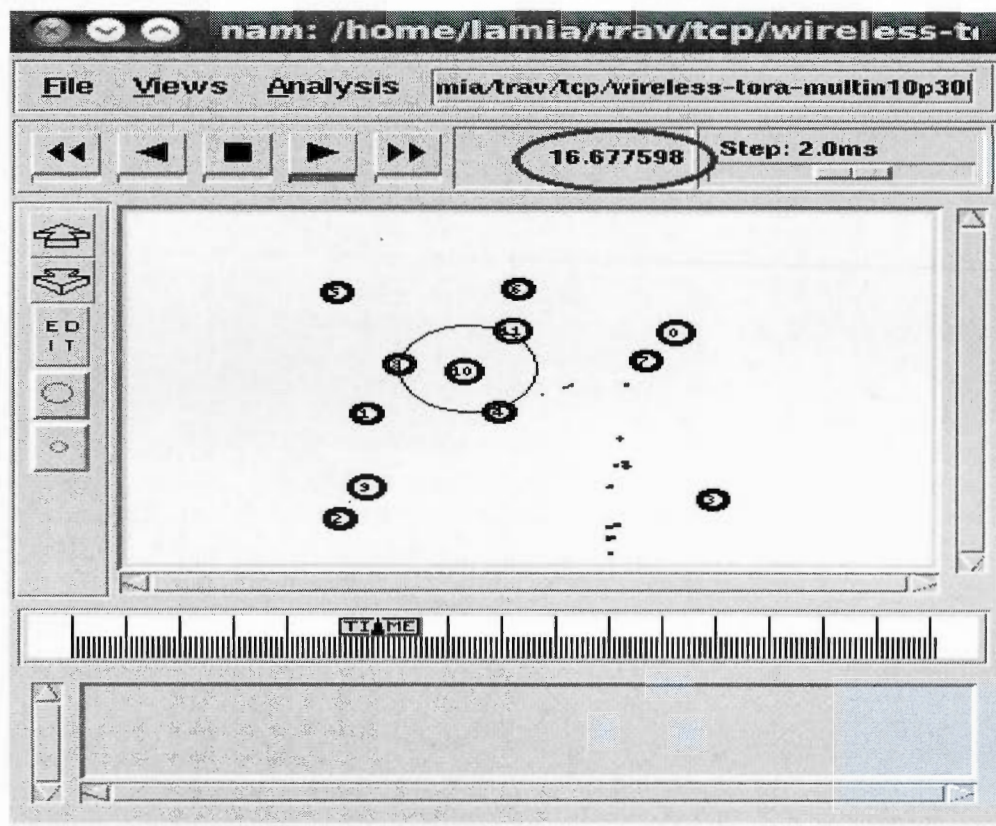


Figure 5.27 Présence de nœuds primaires en zone d'activité de nœuds secondaires

Nous remarquons qu'une perte de paquets est visible sur la figure 5.30 par les points noirs tombants du nœud secondaire n_7 . Le fait que le nœud primaire n_{10} (couleur bleu) commence son processus de découverte de chemin par l'envoi de paquets QRY avec le champ "Réserve" différent de 1, ces paquets sont reçus par les nœuds secondaires (pour les nœuds secondaires, le champ "Réserve" est mis à 1). Cela donne comme résultat un arrêt immédiat de la communication entre n_4 et n_7 . Le chemin n_4 - n_7 est rompu et les paquets envoyés en route sont perdus.

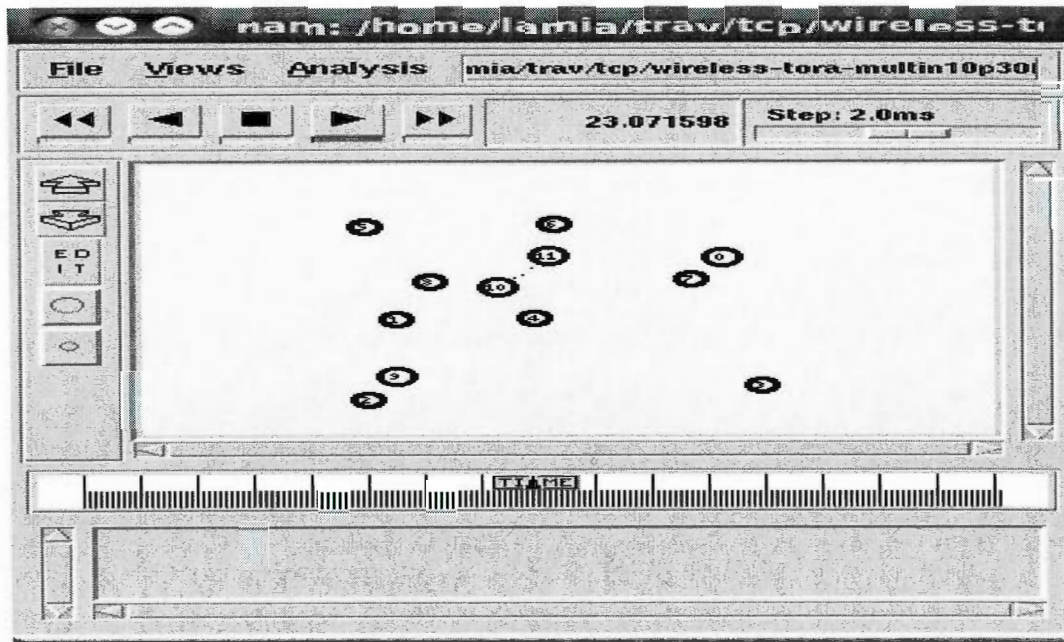


Figure 5.28 Transfert de données entre nœuds primaires et cessation de communication entre nœuds secondaires.

Durant la période de communication entre les nœuds primaires n_{10} et n_{11} , les deux nœuds secondaires essayent de trouver un autre chemin ou de rétablir le premier chemin durant l'absence de trafic des nœuds primaires.

La simulation effectuée a donnée le résultat de débit illustré dans le graphique 5.32.

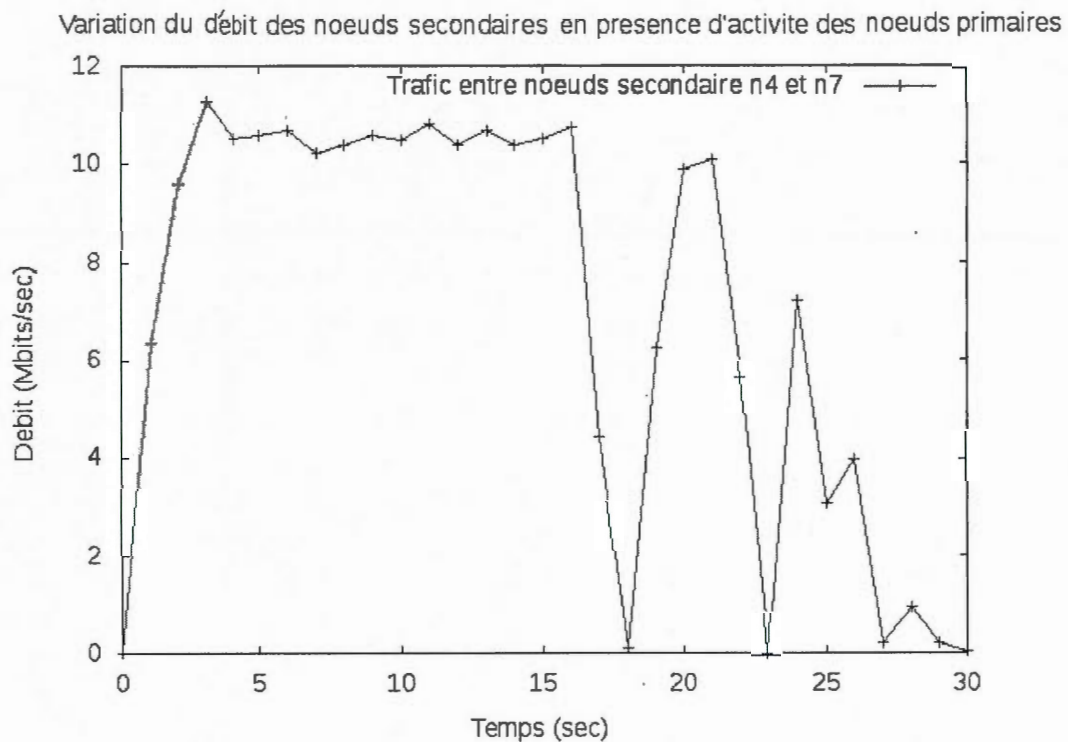


Figure 5.29 Variation du débit entre nœuds secondaires en présence d'activité de nœuds primaires

Nous remarquons que le débit a une légère fluctuation avant l'arrivée des nœuds primaires. À l'instant $t = 18$ s, la valeur de débit chute allant jusqu'à presque s'annuler. Ensuite, la valeur remonte pour atteindre les 10 Mbits/s pour baisser une autre fois. Cette variation allant jusqu'à l'annulation du débit entre n_4 et n_7 c'est-à-dire un arrêt de transmission entre les deux nœuds représente la réaction programmée pour TORA multi-interface afin d'agir en présence de nœuds primaires et de prioriser les activités des nœuds primaires.

Le processus que nous avons programmé afin que TORA multi-interface réagisse à la présence d'activité des nœuds primaires est déclenché. À la réception d'un paquet QRY émis par le nœud primaire n_{10} , les nœuds secondaires transmettent des paquets CLR à leurs voisins directs et interrompent leurs transmissions pour ne pas affecter les transmissions des nœuds

primaires. L'arrêt de transmission de paquets entre nœuds secondaires met à "zéro" le débit de transmission à cet instant. Il est impératif par les nœuds secondaires de chercher un autre chemin pour continuer leurs transmissions. Alors, le déclenchement du processus de maintenance de route de TORA est lancé au voisinage des nœuds où la route est rompue, ce qui explique la remontée des valeurs de débit par la suite. Ainsi les transmissions primaires ne sont pas altérées par les transmissions des nœuds secondaires, ce qui répond aux exigences qu'un protocole de routage dans les RRCs doit assurer.

CHAPITRE VI

CONCLUSION ET PERSPECTIVES

6.1 Conclusion

En conclusion, nous pouvons confirmer qu'il est très important de trouver un protocole de routage capable de contourner tous les obstacles qui peuvent surgir dans un réseau et de répondre aux besoins des utilisateurs qu'il soient primaires ou secondaires sans influencer leurs transmissions. Il est difficile d'adapter un protocole existant en le modifiant afin qu'il réponde aux exigences citées précédemment. L'adaptation que nous proposons exige d'étudier le protocole en détail, d'analyser ses caractéristiques, de prendre en considération son utilisation et les autres plateformes connexes qui réagissent avec lui pour développer et implémenter les modifications nécessaires afin d'obtenir le résultat souhaité. Le protocole TORA est un protocole qui n'a pas été précédemment adapté aux réseaux RRCs ad hoc, à cause de sa complexité et de son emploi d'un autre protocole IMEP pour faire la liaison et le contrôle de l'état des liens.

Ce mémoire aborde le problème de routage dans les réseaux ad hoc à radio cognitives. Nous proposons une solution adaptée faite sur plusieurs étapes. Dans un premier temps, nous nous sommes intéressés aux réseaux à radios cognitives, leur fonctionnement et aux problèmes que nous pouvons rencontrer pour développer une solution efficace de routage. Nous avons introduit quelques termes spécifiques à ce type de réseaux et nous avons défini trois types d'environnement statique, dynamique et opportuniste. Nous nous sommes focalisées sur le troisième type d'environnement. À travers une revue de la littérature, nous avons analysé les différentes solutions et propositions faites jusqu'à présent pour pallier ce problème. Dans une seconde partie, nous avons proposé le protocole de routage réactif TORA avec une adaptation

appropriée comme solution au problème de routage dans les réseaux à radio cognitives. Pour implémenter cette solution nous avons étudié de près le fonctionnement du protocole de routage TORA ainsi que son interaction avec le protocole IMEP. Par la suite, nous avons suggéré les modifications qu'il faut apporter afin de l'adapter aux réseaux à radios cognitives; c'est-à-dire nous avons ajouté le concept multi-interface au protocole de routage TORA. Nous avons ajouté aussi une modification afin de distinguer les paquets des utilisateurs secondaires de ceux des utilisateurs primaires. En cas de dégradation d'une route suite à l'activité d'un nœud primaire, un processus de maintenance de route est lancé par le protocole TORA pour les utilisateurs secondaires et une nouvelle route doit être établie si elle existe. À la fin de ce mémoire, nous avons implémenté notre solution dans le simulateur NS-2 pour évaluer ses performances. Les résultats obtenus ont démontré que la solution proposée donne une amélioration significative au niveau de débit, de taux de livraison et de délai de bout en bout; ce qui encourage à essayer de l'implémenter sur une plateforme réelle.

6.2 Perspectives

L'apport d'une amélioration à TORA pour le rendre plus efficace en ce qui concerne la question de débit et de taux de livraison de paquets démontre que notre solution donne une satisfaction pour l'employer dans des réseaux réels et de voir cette satisfaction au niveau des utilisateurs. Nous remarquons aussi certains points à ajouter dans le futur comme amélioration et perfectionnement que nous pouvons porter à notre solution dans le cadre des perspectives que nous proposons. Certes la solution proposée est un début d'adaptation d'un protocole classique mature qui peut répondre aux besoins de routage dans les réseaux RRCs ad hoc, sauf qu'il sera très intéressant d'ajouter quelques améliorations que nous voyons primordiales afin de le peaufiner pour un usage performant.

Nous proposons de changer la métrique de routage utilisée dans ce protocole de routage par une métrique plus significative. Plusieurs métriques ont été étudiées et appliquées sur les réseaux RRCs. Nous citons la métrique ETX (en anglais *Expected Transmission Count*) (S. Yaye, 2011). Cette métrique calcule le nombre de transmission et retransmission pour transférer un paquet sur un lien. Aussi la métrique ETT (en anglais *Expected*

Transmission Time) est une amélioration de ETX. Elle prévoit le temps qu'un paquet prend pour parcourir un lien. Sauf que ces deux métriques ne sont pas dédiées aux réseaux multi-interface multi-canal, ce qui nous pousse à chercher des métriques qui répond le mieux à notre besoin. La métrique WCETT (en anglais *Weighted Cumulative Expected Transmission Count*) est une amélioration de ETX pour les réseaux multi-interface multi-canal. Elle permet d'éviter l'interférence due à l'inter-flux des réseaux multi-canal. Aussi une nouvelle métrique MBP (en anglais *Metric Based on Probabilities*) basée sur le principe des trois métriques citées précédemment offre une amélioration d'une combinaison des trois. La métrique MBP prend en considération le taux de perte, les interférences et la qualité des liens. Sur cette base, cette métrique calcule la probabilité qu'une transmission sur un chemin soit réussie (S. Yaye, 2011).

De ce fait il sera très avantageux à notre solution d'employer la nouvelle métrique pour assurer des chemins offrant une grande réussite de transmission et fiabilité. Aussi, nous proposons de perfectionner le processus d'optimisation de chemin du protocole TORA, car ce processus est considéré comme secondaire et le plus important dans TORA est la découverte d'un chemin. L'optimisation de chemin nous fera gagner plus de temps de transmission entre les nœuds secondaires et elle réduira le délai de bout en bout et les interruptions de transfert dues à la présence de nœuds primaires.

Une dernière perspective à proposer sera d'inclure des échanges inter-couches entre la couche MAC et la couche contrôle de liaison physique pour choisir les canaux les plus stables durant un temps de transmission. Ces canaux seront favorisés pour un premier établissement de communication. Avec ces améliorations, nous achevons ce travail pour penser à étudier de près leur mise en œuvre.

BIBLIOGRAPHIE

Angela Sara Cacciapuoti, Cosimo Calcagone et Marcello Caleffi (2010). CAODV : Routing in mobile ad hoc cognitive radio networks. IEEE conference publication 2010. Pages 1-5

Ashwin Sampath Lei Yang Lili Cao Haitao Zheng et Ben Y. Zhaou. High Throughput Spectrum-aware Routing for Cognitive Radio Networks. IEEE crowncom 2008.

FCC (2012). <http://www.fcc.gov/pshs/techtopics/tech-interop.html>

Guo-Mei Zhu, Ian F. Akyildiz et Geng-Sheng (G.S.) Kuo (2008). STOD-RP: A Spectrum-Tree Based On-Demand Routing Protocol for Multi-Hop Cognitive Radio Networks. Global Telecommunication Conference 2008. Pages 1-5

Hicham Khalifé, Naceur Malouch et Serge Fdida(2009). Multihop Cognitive Radio Networks: To Route or Not to Route. IEEE Network 2009. Pages 20-25

Hicham Khalifé et Naceur Malouch (2009). Routage dans les réseaux radios Cognitifs. 10^{ème} Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications Avril 2009.

Insa Ngom et Louis Diouf (2008). La radio cognitive 2008.

Ioannis Pefkianakis, Starsky H.Y Wong et Songwu Lu (2008). Samer : Spectrum Aware Mesh Routing in Cognitive Radio Networks. 3rd IEEE Symposium 2008. Pages 1-5

Jian Chen, Hewu Li et Jianping Wu (2010). WHAT: A Novel Routing Metric for Multi-Hop Cognitive Wireless Networks. Wireless and Optical communication conference IEEE 2010. Pages 1-6

K.R. Chowdhury et M.D. Felice (2009). Search A routing protocol for mobile cognitive radio ad hoc networks. IEEE SARNOFF 2009. Pages 1-6

Kaushik R. Chowdhury Ian et F. Akyildiz (2010). CRP Routing Protocol for Cognitive Radio Ad hoc Networks. IEEE Journal 2010. Pages 794 – 804

Harmider S. Binda, Sunil K. Maakar et A. L. Sangal (2010). Performance Evaluation of Two Reactive Routing Protocols of MANET using Group Mobility Model. IJCSI International Journal of computer Science. Vol 7, Issues 3, N°10 2010. Pages 38-43

Lei Ding, Tommaso Melodia, Stella Batalama et Michael J. Medley (2010). Cross-Layer Routing and Dynamique Spectrum Allocation in cognitive Ad hoc Networks. IEEE Transaction on 2010. Pages 1969-1979

Liljana Gavrilovska et Vladimir Atanasovski (2011). Spectrum Sensing Framework for Cognitive Radio Networks. Wireless Pers Commun Springer 2011.

Matteo Cesana, Francesca Cuomo et Eylem Ekici (2010). Routing in cognitive radio networks: Challenges and solutions. ELSEVIER 2010

Pingnan Lee et Gengyu Wei (2009). NS2 Model for Cognitive Radio Networks Routing. Computer Netowrk and Multimedia Technology 2009. Pages 1-4

Quansheng Guan, F. Richard et Yu Shengming Jiang (2012). Protocol Cognitive Topology Control and routing in Mobile Ad hoc Networks with Cognitive Radios. IEEE Wireless communication 2012. Pages 74-79

Ramon Aguero Calvo et Jesus Perez Campo (2007). Adding Multiples Interface Support in NS2. 2007

Suyang Ju Joseph B. Evans (2009). Spectrum-Aware Routing Protocol for Cognitive Ad hoc Networks. IEEE Globecom 2009 proceedings. Pages 1-6

The network simulator NS2 (2012). <http://www.isi.edu/nsnam/ns/> Novembre 2011

Venkaiahnaidu.A, B.RameshBabu, Sk.MansoorRahaman et K.RajasekharaRao (2012). A New OLSR Routing Protocol in Cognitive Wireless Mesh Networks. International Journal of Modern Engineering Research Jan-Feb 2012. Pages 496-498

Vincent Park et Scott Corson (1997). An Internet MANET Encapsulation Protocol (IMEP). Novembre 2011.

Vincent D. Park et M. Scott Corson (1997). Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification. Novembre 2011.

Yang Xu, Min Sheng et Yan Zhang (2010). Traffic-aware Routing Protocol for Cognitive Network. IEEE 2010. Pages 1-5

Sarr Yate (2011). Métriques de routage dans les réseaux maillés sans fil. Mémoire de maîtrise en sciences en informatique 2011. Université de Montréal. Avril 2012.

Zhenhui Zhai, Yong Zhang, Mei Song et Guangquan Chen (2010). A reliable and adaptative AODV protocol based on Cognitive routing for Ad hoc Networks. The 21th International Conference. IEEE 2010. Pages 1307-1310